



ELSEVIER

Contents lists available at ScienceDirect

Control Engineering Practice

journal homepage: www.elsevier.com/locate/conengprac

Bridging semantics, control specifications and assessment: A library for scalable demand flexibility controls

Flavia de Andrade Pereira ^{id} a,b,c, Marco Pritoni ^{id} c,*, Armando Casillas ^{id} c, Jessica Granderson ^{id} c, Lazlo Paul ^{id} c, Anand Prakash ^{id} c, Conor Shaw ^{id} a, Dimitrios Rovas ^{id} b, Susana Martin-Toral ^{id} d, Donal Finn ^{id} a, James O'Donnell ^{id} a,e

^a School of Mechanical & Materials Engineering and UCD Energy Institute, University College Dublin, Dublin, Ireland

^b University College London, London, United Kingdom

^c Lawrence Berkeley National Laboratory, Berkeley, United States of America

^d CARTIF Technology Centre, Energy Division, Valladolid, Spain

^e School of Engineering, University of Limerick, Limerick, Ireland

ARTICLE INFO

Keywords:

Demand flexibility
Controls library
Portability
Semantic models
Scalability

ABSTRACT

There is growing recognition that Demand Flexibility (DF) can play a major role in enhancing grid reliability, with building control applications emerging as key enablers for DF. However, the traditional approach to deploying new control applications in buildings, including those for DF, remains largely manual and tailored to individual buildings, making it difficult to scale. While research efforts have explored semantics-driven portability, DF controls specification, and assessment approaches, these initiatives are fragmented and limited in scope. This paper proposes a novel methodology, grounded in design science research, to integrate these elements and create a comprehensive DF controls library for both industry and academia. This approach is applied to develop the Demand FLEXibility controls LIBrary using Semantics (DFLEXLIBS), an extensible open-source library that provides DF controls for HVAC systems in Python. DFLEXLIBS enables portable, easy-to-deploy controls that abstract building-specific data points, facilitating assessment across diverse buildings. DFLEXLIBS features nine different control applications, and it is successfully implemented and tested across four virtual and two real buildings, bridging the gap between semantics-driven portability, DF controls specification, and rigorous performance assessment. Its benefits are measured by a reusability ratio greater than 90% and a functional overlap ratio of around 70% for the most common functions used in the library, significantly reducing time for deploying new controls.

1. Introduction

The growing use of intermittent renewable sources is widening the gap between energy supply and demand, raising the risk of energy waste through curtailment and posing challenges for grid stability (U.S. Energy Information Administration, 2023). This has made investments in the grid, especially in Demand Flexibility (DF), more pressing than ever (International Energy Agency, 2024). DF refers to the ability of consumers to dynamically manage their energy demand and/or onsite generation to support the grid according to weather conditions, grid requirements, and user needs (e.g., thermal comfort, productivity) (Li et al., 2021). Due to their share in global demand, buildings play a pivotal role in

offering DF, accounting for more than half of the global DF targets by 2030 (International Energy Agency, 2023). Among DF applications for buildings, leveraging passive thermal storage via building thermal mass is highly impactful and cost-effective compared to alternatives such as behind-the-meter batteries (Liu et al., 2023). Yet, the implementation of such a control approach in heterogeneous buildings still lacks scalability and generalizability, especially in large commercial buildings.

Key barriers to controls scalability include i) the ambiguity of traditional natural-language specifications, often leading to numerous control implementation variants (Faulkner et al., 2023); and ii) the interoperability challenges arising from the unique nature of each building (its heterogeneous data sources and inconsistent naming conventions

* Corresponding author.

E-mail addresses: flavia.pereira@ucl.ac.uk (F. de Andrade Pereira), mpritoni@lbl.gov (M. Pritoni), acasillas@lbl.gov (A. Casillas), jgranderson@lbl.gov (J. Granderson), LPaul@lbl.gov (L. Paul), akprakash@lbl.gov (A. Prakash), conor.shaw@ucdconnect.ie (C. Shaw), d.rovas@ucl.ac.uk (D. Rovas), susmar@cartif.es (S. Martin-Toral), donal.finn@ucd.ie (D. Finn), james.odonnell@ucd.ie (J. O'Donnell).

<https://doi.org/10.1016/j.conengprac.2026.106899>

Received 25 August 2025; Received in revised form 20 January 2026; Accepted 4 March 2026

Available online 10 March 2026

0967-0661/© 2026 The Author(s). Published by Elsevier Ltd. This is an open access article under the CC BY-NC-ND license (<http://creativecommons.org/licenses/by-nc-nd/4.0/>).

within Building Automation Systems (BAS)), typically resulting in labor-intensive and bespoke onboarding processes to discover and map building data points (Chamari et al., 2023b; Duarte Roa et al., 2023; Waterworth et al., 2021). Together, these have led to control applications that are tightly hard-coded to individual buildings and have driven the creation of proprietary control libraries, often relying on custom programming languages (Schraven et al., 2019), limiting portability and reuse.

Semantic models based on ontologies, such as Brick,¹ SAREF,² and the upcoming ASHRAE 223 standard,³ have been recognized as key enablers to enhancing interoperability and portability of building applications (de Andrade et al., 2025; Li & Hong, 2022; Li et al., 2021). Semantic models employ logic notations and common concepts from ontologies to enable shared context and meaning across data from various sources (Balaji et al., 2018; Bergmann et al., 2020; Roth et al., 2022). By enabling building systems to be described in a standardised manner, these models allow applications to consistently map their data needs across diverse buildings, instead of hard-coding references to specific points from the underlying data sources (Bennani et al., 2021).

Semantics-driven frameworks have recently been proposed to enable analytics and controls with consistent data access across buildings, eliminating the need to develop custom applications (de Andrade et al., 2025, 2024; Paul et al., 2024). Examples include analytics-focused frameworks such as Mortar (Fierro et al., 2019), Energon (He et al., 2021), SeeQ (Mavrokapnidis et al., 2023a,b), and others (Chiosa et al., 2024; Kucera & Pitner, 2018; Tomasevic et al., 2015; Xu et al., 2024), as well as a smaller number of controls-focused frameworks (Chamari et al., 2025; de Andrade et al., 2024; Paul et al., 2024; Wan et al., 2025). While some of these efforts show promise in streamlining the setup of DF controls, and many leverage open programming languages, they fall short in other critical aspects. They do not follow best-practice sequences in control specifications and lack methods to assess how well portable controls perform across different buildings. These are important to increase the reusability of controls, and because portability alone does not necessarily ensure the effectiveness or suitability of controls for all buildings, performance assessments are necessary.

This paper aims to propose a novel methodology to guide the design, implementation, and evaluation of a library of reference and portable supervisory DF control applications. The work outlines key principles and requirements for creating such a library and introduces the Demand FLEXibility controls LIBrary using Semantics (DFLEXLIBS) as a prototype. DFLEXLIBS is an open-source, Python-based library that provides a holistic integration from several previously fragmented research areas, including controls specification, performance assessment, and semantics-driven portability of building applications. It is populated with the best-practice DF control specifications from Granderson et al. (2025), Granderson and House (2023) and integrated with the Energy Flexibility (EF) Key Performance Indicators (KPI) Python package from Johra et al. (2023). Built on our established semantics-driven framework (de Andrade et al., 2024), DFLEXLIBS abstracts building-specific data points, allowing a single code base to be easily applied across various buildings. This not only makes it easier to reuse (or port) controls, but also enables quick testing and performance evaluation across a variety of conditions, including different buildings, environmental factors, constraints, and grid signals.

With this aim, the remainder of this paper is organized as follows. Section 2 includes a background, summarizes research gaps, and states the contributions of this paper. Section 3 describes the methodology devised to guide the design, implementation, and evaluation of a library of portable controls. Section 4 presents DFLEXLIBS components following the proposed methodology. Section 5 discusses the main findings.

Finally, Section 6 outlines concluding remarks and future research directions.

2. Background and contributions

To explore how controls' best-practice specifications and assessment can be incorporated into the development of a semantics-driven library of portable DF controls, this section reviews current research on DF controls specification, performance assessments, and existing libraries of controls. It also identifies their key research gaps and outlines the specific contributions of this paper, connecting these fragmented research areas.

2.1. DF controls specification

Several national and international standards and guides are available as reference documents for specifying, developing, and configuring control applications. Some examples include ISO 16484-3 (Building automation and control systems, BACS) and the Application Guide (AG 7/98) from Building Services Research and Information Association (BSRIA) (Martin et al., 1998) for low-level BAS control sequences, as well as the ASHRAE Guideline 36 (G36) for best-in-class HVAC control sequences (ASHRAE Guideline 36-2018 - High-Performance Sequences of Operation for HVAC Systems, 2021). Although G36 includes some strategies suitable for DF, none of these standards and guides particularly focus on DF controls.

To streamline the design and implementation of DF controls for HVAC systems in commercial buildings, Motegi et al (2007) developed a guide intended for use by building professionals such as facility managers, building owners, and control developers (Motegi et al., 2007). This guide categorizes and describes various controls, providing technical details for implementing them, along with recommended sequences of operation. However, because such sequences rely solely on natural language descriptions, without any executable specifications or even pseudocode, it is difficult to understand their underlying logic and ensure consistent implementations.

Introducing new specifications for best-practice DF controls of HVAC systems in commercial buildings, the studies of Granderson et al (2023) and (2024) aim to partially address this gap (Granderson et al., 2025; Granderson & House, 2023). In its first work, Granderson provides an implementation guide listing a number of control strategies that enable load shedding, shifting, and demand-limiting. To enhance understanding and scalability, in its second work, Granderson documents the specifications of these strategies in pseudocode, validating some of them in real-world buildings using different Energy Management Information Systems (EMIS) vendors. While this research represents a significant advancement by providing generalizable control sequences and reducing the effort required for their specification, it falls short in offering open-source implementation codes and assessing their performance across multiple buildings.

Making these resources openly accessible and vendor-independent would prevent redundant efforts in converting pseudocode into implementation code and enable rapid virtual vetting of the controls across various building system configurations. In addition, understanding the performance differences among buildings and the primary factors influencing them would help control developers gain confidence before implementing the controls in real scenarios.

2.2. DF controls assessment

Proven performance is especially important for DF controls, not only to establish trust among control developers, building owners, and operators, but also because some contracts may impose penalties for under-performance or revenue losses due to overly conservative estimates (Liu et al., 2022). However, due to the inherent variability and uncertainty of building energy flexibility across different contexts, guaranteeing the

¹ <https://brickschema.org/ontology>.

² <https://saref.etsi.org/core/v3.1.1/>.

³ <https://docs.open223.info/intro.html>.

performance of DF controls upfront without sufficient testing is challenging (Li et al., 2023).

While studies that propose control specifications often do not include methods to evaluate the controls, another line of research has focused on approaches to investigate and quantify the variability and uncertainty of DF controls' performance across different contexts. Common influential factors encountered include building characteristics (e.g., type, age, and operating hours), single-event attributes (e.g., event duration, timing, baseline controls, and weather conditions), and stochastic factors (e.g., occupancy levels, internal loads, and commissioning issues) (Liu et al., 2023). For instance, the research conducted by Yin et al. (2023) suggests that large office buildings tend to demonstrate the least reduction in energy demand units (W/m^2) during shedding events compared to small and medium office buildings. This is likely due to the superior efficiency of their HVAC systems, resulting in a smaller decrease in kilowatt demand. Similarly, a study by Luo et al. (2022) found comparable results among buildings of different vintages, indicating that newer buildings exhibit greater efficiency and experience a smaller decrease in demand compared to older ones.

Although these analyses offer valuable insights into estimating DF potential and understanding performance variability (Yin et al., 2016), they do not fully capture the complexity of real-world controls and building dynamics. They often rely on simulating controls with schedules and setpoints applied to generic prototype buildings, modelled with tools such as DOE-2, EnergyPlus, and TRNSYS. While robust in modeling envelope physics and various HVAC systems, these tools are primarily designed for annual energy use calculations and lack support for simulating controls in a realistic and explicit manner (i.e., incorporating baseline controls and enabling overwriting supervisory control signals based on dynamic logic) (Wetter et al., 2020; Zhang et al., 2022). As such, their practical application in a pipeline that supports the assessment of control applications, which can be potentially exported from simulation and reused in real buildings, is limited.

Some studies, while not explicitly focusing on DF, have explored the use of the Modelica language to facilitate more realistic simulations of controls integrated with high-fidelity building models. One example of these efforts is the Building Optimization Testing Framework (BOPTTEST), a standardized benchmarking platform that allows testing controls using realistically modeled buildings in Modelica (Blum et al., 2021). It provides an API similar to a BAS/EMIS, which could be powerful to address the limitations of current studies testing and assessing DF controls (Roth et al., 2022). Other efforts in this area proposed the development of libraries with controls that can also be tested in Modelica models. These are further discussed in the following subsection.

2.3. Existing libraries of controls

While no library of DF controls has so far been integrated with BOPTTEST for testing and assessing more realistic simulation of controls, some efforts have proposed the development of libraries with controls that can also be tested in Modelica building models. These libraries aim to reduce the need to create controls from scratch or convert predefined textual specifications into reusable code syntax, allowing the simulation of controls against Modelica models. To better understand how they work, we reviewed the open-source efforts by AixOCAT (Schraven et al., 2019), BuildingControlLib (Schneider et al., 2017), AixLib (Maier et al., 2023), BESMod (Wüllhorst et al., 2023), and the Modelica Buildings library (Wetter et al., 2014). Table 1 provides a comparison of these libraries, highlighting criteria relevant to the scope of this paper, which are elaborated on in the following paragraphs.

To date, most libraries mainly focus on low-level control functions rather than **supervisory controls**. While some, such as the Modelica Buildings Library, include supervisory controls like the G36 control sequences, they do not yet support **DF controls**. This is likely because G36 sequences are well-established and standardized by ASHRAE, whereas no similar standardization effort has been made for DF controls.

Table 1

Comparison of reviewed open-source controls' libraries. ++ criteria fulfilled, + partially fulfilled and - not fulfilled. ACT: AixOCAT (Schraven et al., 2019), BCL: BuildingControlLib (Schneider et al., 2017), AL: AixLib (Maier et al., 2023), BES: BESMod (Wüllhorst et al., 2023), MBL: Modelica Buildings library (Wetter et al., 2014).

| Criteria | ACT | BCL | AL | BES | MBL |
|------------------------|-----|-----|----|-----|-----|
| Supervisory controls | - | ++ | - | ++ | ++ |
| DF controls | - | - | - | - | - |
| Modular design | ++ | ++ | ++ | ++ | ++ |
| Semantic aspects | - | + | - | - | + |
| Multi-buildings assess | + | + | + | + | + |
| Real testing | + | - | - | - | + |

All libraries adopt a **modular design approach**, offering a set of common pre-built control functions, often called functional blocks, that can be combined to create control applications tailored to specific system configurations. This modular design ensures code extensibility and reusability, facilitating more efficient control design, implementation, and maintenance, ultimately reducing engineering effort.

Aiming to enhance interoperability, some of these libraries have also incorporated **semantic aspects**. BuildingControlLib (Schneider et al., 2017) introduces a naming convention to specify exchange variables from the control functions. The primary goal is to ensure seamless compatibility among interconnected functions, achievable only when their inputs and outputs are aligned in unit, quantity, point type, and direction (in/out). Despite its potential to enable a more cohesive and automated binding of variables across different functions, the study suggests that the proposed custom naming convention was cumbersome to maintain and susceptible to errors. In response to these challenges and to streamline the configuration (instantiation) of controls across BAS/EMIS platforms, ongoing work is being conducted to support the Modelica Buildings library in annotating controls' inputs and outputs using standardized concepts from the Brick ontology (Roth et al., 2022). While BESMod acknowledges the potential of a similar approach (Wüllhorst et al., 2023), no practical demonstrations have been provided yet.

By integrating with Modelica models for buildings and HVAC systems, the controls provided by these libraries can be tested through dynamic simulation and offer **multi-building assessment** realistically. However, to date, none of them have presented a thorough performance analysis of given controls tested against different buildings.

Finally, since most of these libraries do not provide a connection between their simulated controls and actual buildings' BAS/EMIS, they lack support for **real building testing**. Two exceptions are the AixOCAT and the Modelica Buildings library. Apart from offering interfaces for Modelica models, AixOCAT provides an object-oriented programming style compatible with real-world Programmable Logic Controllers (PLCs). However, there are no demonstration studies on it. In the case of the Modelica Buildings library, it has a set of tools within the OpenBuildingsControl project that facilitate a control delivery workflow from Modelica simulation to real buildings (Wetter et al., 2020, 2022). Among these tools, there is a Modelica-json converter (Wetter et al., 2021) that exports the Modelica controls specified in Control Description Language (CDL) (Wetter et al., 2018), an upcoming ASHRAE Standard (231), into a vendor-independent intermediate format in JSON-LD called Control eXchange Format (CXF). Translator tools can then use this file in CXF to convert the controls' specifications into specific platform languages such as Automated Logic's EIKON (Wetter et al., 2022). Using these tools, controls can be customized, tested with energy models, exported, and translated into codes compatible with real buildings' BAS/EMIS. Despite increased automation, potential drawbacks include the need for translations into compatible programming languages, a lack of real-world validations, and the lack of familiarity of most practitioners with the Modelica language.

2.4. Research gaps and contributions

Overall, although many studies have explored different aspects of DF applications, such as their portability, specification, and assessment, existing works remain highly fragmented, and none have integrated all these elements comprehensively. Current semantics-driven approaches for the portability of DF controls do not rely on well-defined specifications and are not linked with proper assessment tools. Most efforts proposing control specifications do not focus on DF, and those that do lack implementation codes and testing across diverse building types. While some works have explored the performance of DF controls in multiple buildings, they usually rely on whole-building simulation tools that use generic control strategies and prototype models. This limits the possibility of realistically testing the controls and making them easily portable to real environments. Finally, while efforts have been made to develop libraries of more realistically simulated controls and software tools to provide end-to-end control delivery workflows, none have specifically focused on DF. They also require expertise in the Modelica language.

Motivated by these limitations, our research aims to develop a novel library for supervisory DF controls, supporting control developers from academia and industry. By leveraging our previous semantics-driven framework (de Andrade et al., 2024), and its success in achieving the portability of controls, we hypothesize that:

"A semantics-driven library of best-practices DF controls, implemented in Python and integrated with a realistic testing environment, could enable the portability of DF controls at scale and with minimal effort while simplifying their onboarding and performance assessment across buildings."

To achieve this aim and verify the hypothesis, this paper focuses on the following contributions:

- identifies the essential design principles and requirements for developing a library of reference and portable DF control applications.
- proposes an open-source prototype of the library, DFLEXLIBS,⁴ which supports the controls' portability and assessment across different simulated and real buildings.
- evaluates the library's benefits through novel metrics and insights to determine whether it can reduce controls' development and deployment efforts.

3. Methodology

We adopted a design science research methodology to verify the stated hypothesis. As shown in Fig. 1, the methodology is based on the approach of Holmström et al. (2009), which aims to bridge practice (exploration) with theory (explanation) types of research. The approach is inherently iterative and consists of three main phases: solution incubation, which focuses on developing an initial solution design; solution refinement, which involves prototyping and improving the initial design; and explanation, which provides theoretical insights and establishes the solution's practical relevance. The explanation phase *"contributes to the generalizability of the results [...] not in the statistical sense, but rather in the theoretical sense"* (Holmström et al., 2009). In other words, it validates the theoretical merit of an idea.

In the incubation phase, we identify the problem (research gap) and hypothesis, and propose and justify an initial library design based on background literature. The problem and research gap are introduced in Section 2, and the preliminary design is presented in Sections 3.1 and 3.2. Section 3.1 specifies design principles and requirements, defines the intended users, functionalities, constraints, and potential user interactions with the library. Meanwhile, Section 3.2 establishes metrics to evaluate the library's benefits (i.e., its ability to reduce the effort needed for control development and deployment). The refinement phase involves implementing a prototype to refine the initial design. This phase,

presented in Section 4, introduces DFLEXLIBS as the prototype. It details the design, implementation (through coding, testing, and assessment of reference control sequences), and evaluation of DFLEXLIBS. Finally, the explanation phase, presented in Section 5, discusses the hypothesis, providing insights into the effectiveness and usefulness of the DFLEXLIBS prototype and determining whether creating a library of portable controls is a worthwhile endeavor.

3.1. Library design

Effectively designing software technologies requires understanding the aims of potential users and use cases, gathering and analyzing requirements to meet these aims, and defining design principles to guide a robust, scalable, and maintainable implementation. The library presented in this paper aims to support control developers in industry and academia who seek to streamline the implementation, assessment, and portability of well-defined DF control applications across various buildings. To achieve this, we propose a library that provides readily available portable and reference control applications complemented with comprehensive descriptions and software artifacts for understanding, evaluating, and reusing them. For developing such a library, we introduce a set of design principles derived from a combination of theory-based knowledge, drawn from a comprehensive review of existing literature, and our expertise and insights gained while demonstrating this work. We also define requirements that are both functional (what the library should do) and non-functional (what the library constraints and ecosystem are).

3.1.1. Design principles

The library design should adhere to the following principles, inspired by the quality attributes proposed in Bartusiak et al. (2022) for building an open, interoperable architecture to support reusable control applications.

- **Modularity:** to be easily shared and extended across various use cases, the functions making up control applications must be modular, as proposed in Martin et al. (1998), Schneider et al. (2017), Schraven et al. (2019), Wetter et al. (2022).
- **Generalizability:** to be portable across heterogeneous buildings, the control applications within the library must be generalizable (i.e., abstracted from specific building data points, flexible to available data, and adaptable to dynamic operating conditions), as introduced in de Andrade et al. (2024).
- **Scalability:** to enable the replicability of the library's control applications across different buildings, they must be based on open, best practice specifications, such as those proposed in Granderson et al. (2025), Granderson and House (2023).
- **Openness:** to prevent vendor lock-in, the controls should rely on an open-source and vendor-independent executable programming language, as suggested in Schraven et al. (2019).
- **Discoverability:** to improve user experience, the controls need to be easily discoverable, as recommended in Rogers et al. (2023).

3.1.2. Functional requirements

Inspired by the requirements of existing control libraries (Schneider et al., 2017; Schraven et al., 2019; Wetter et al., 2014; Wüllhorst et al., 2023), and given that the main functionality of the library is to support the development and deployment of controls that can be easily tested, assessed, and reused across different buildings, the library's underlying structure should enable users to:

- Create new or reuse and customize existing control functions to compose control applications that can be exported as implementation codes.
- Create software artifacts to support the configuration of the controls. These include semantic queries, configuration files, and interfaces to BAS/EMIS and simulation platforms that allow the controls to

⁴ <https://github.com/LBNL-ETA/DFLEXLIBS>

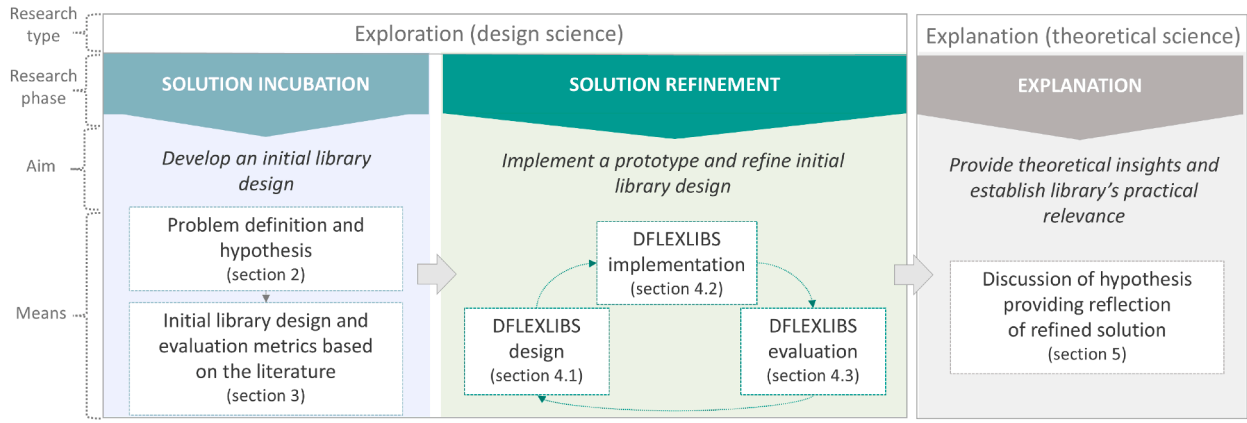


Fig. 1. Overarching design science research-inspired methodology to guide the design, implementation, and evaluation of a library of portable DF control applications.

be instantiated and customized in given buildings, as described in de Andrade et al. (2024).

- Easily connect, via appropriate interfaces, to simulation environments for testing controls and their portability before deployment, as well as to BAS/EMIS platforms for their implementation in real buildings.
- Seamlessly integrate with analytic applications, such as KPI-based, to assess the performance of controls in both simulated and real buildings.

In addition, following common practices for specifying controls, as outlined in standards such as ASHRAE Guideline 36-2018 - High-Performance Sequences of Operation for HVAC Systems (2021), Building automation and control systems (BACS), Martin et al. (1998), and aligning with the DF controls categorization suggested in several academic studies (Fu et al., 2022; Motegi et al., 2007; Santos et al., 2019), the library, if equipped with a Graphical User Interface (GUI) that adheres to the discoverability principle, should enable users to:

- Identify available controls by i) searching keywords; ii) navigating via well-defined categories such as types of buildings and systems (e.g., typologies like residential or commercial buildings and their employed systems such as HVAC types), target levels (e.g., zone-level, plant-level), and/or load shapes (e.g., shedding, shifting); or iii) using semantic validation mechanisms which allow to filter controls based on their suitability for given buildings (de Andrade et al., 2024; Fierro et al., 2022)
- View available controls' descriptions, flow charts, and adopted algorithm types (e.g., rule-based).
- Access available controls' data requirements defined as lists of input/output data points, including direct sensor readings, user-defined parameters, default values, and/or computations derived from the others.
- Review results from performance assessments of available controls tested across various buildings.

3.1.3. Non-functional requirements

To enable control applications exported from the library to seamlessly interact with both simulation and real building ecosystems (platforms), they should be accompanied by software artifacts that configure the controls and enable effortless data exchange through the appropriate interfaces. Fig. 2 presents a generic software architecture, building on concepts explored in Blum et al. (2021), Chamari et al. (2023a), de Andrade et al. (2024), Pauwels and Fierro (2022), that illustrates how we envision the integration of DF controls exported from the library with BAS/EMIS and simulation platforms through these supporting artifacts. It includes: i) a data ingestion layer that interacts directly with deployed/simulated devices and external data sources; ii) an abstraction layer that provides standardized interfaces from these devices

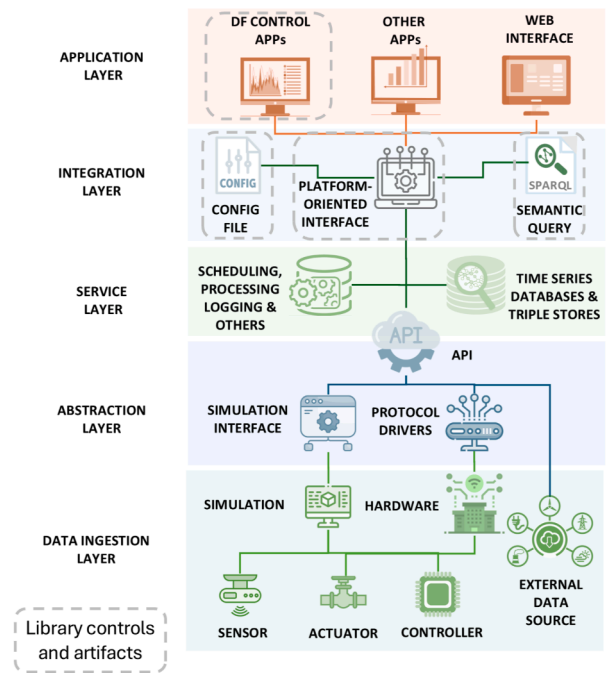


Fig. 2. Connection between exported software artifacts from the library with common elements from BAS/EMIS and simulation software architectures.

and data sources to the rest of the platform; iii) a service layer that offers scheduling, data processing, logging, storage and other essential functionalities; iv) an integration layer that includes the library's software artifacts to bind its exported control applications to the specific buildings using platform-oriented interfaces that can self- and custom-configure the controls, via semantic queries (SPARQL⁵) and configuration files, respectively. Finally, an application layer on top that provides a run-time environment for executing the exported control applications on the given platform.

3.2. Library evaluation

As the final step of this methodology, we propose two main metrics to evaluate a library's effectiveness in supporting the portability of controls and minimizing their development and deployment efforts. The first metric is called Reusability Ratio (RR), and was inspired by other

⁵ <https://www.w3.org/TR/sparql11-query/>

studies that use Lines Of Code (LOC) to evaluate the portability of building applications (Mavrokapnidis et al., 2023a) and mobile applications (Tung et al., 2018). The RR measures the relative change in LOC between an existing application on a reference building and the version adapted for a new building. A higher RR indicates greater reusability and less effort required for adaptation. RR is expressed as a percentage (%) and calculated as:

$$RR = 100 \left(1 - \frac{LOC_{\text{adapted}}}{LOC_{\text{reference}}} \right) \quad (1)$$

where $LOC_{\text{reference}}$ is the total lines of code in the existing application on a reference building, and LOC_{adapted} refers to the lines of code that were changed, added, or removed to adapt it to a new building.

The second proposed metric, called Function Overlap Rate (FOR), quantifies the overlap of pre-built control functions in different control applications. This metric is inspired by the reusability assessment methodologies previously introduced for component-based model-driven software development (Umran Alrubae et al., 2020). A higher FOR, indicated by their greater overlap among the controls, corresponds to higher reusability and less effort required to create new controls. FOR is expressed as a percentage (%) and calculated as:

$$FOR = \frac{U_f}{N} \quad (2)$$

where N is the total number of control applications being evaluated, U_f is the number of control applications that use the function f .

Overall, with these generic design criteria, requirements, and evaluation metrics in place, we believe this methodology can support the creation and validation of a library that works across multiple DF controls and both simulation and real-building platforms. The next section demonstrates this with a prototype library designed, implemented, and evaluated according to the proposed methodology.

4. DFLEXLIBS prototype

In line with the design principles and requirements outlined in the previous section, we have developed a library called DFLEXLIBS to showcase an implementation of the methodology. Fig. 3 illustrates the main components and software artifacts of the library, which facilitate coding, testing, assessing, and exporting the DF control applications for HVAC systems. These components include: 1) DF control applications as implementation codes; 2) interfaces to virtual, realistically modeled buildings in Modelica (via BOPTTEST Blum et al., 2021) and to real buildings (via VOLTTRON Katipamula et al., 2016) for testing and implementing the controls; 3) DF analytic applications that evaluate the behavior and performance of the controls in both simulated and real buildings; and 4) a semantics-driven framework based on previous research (de Andrade et al., 2024), which serves as the core of DFLEXLIBS, enabling the DF control and analytic applications to be easily portable across heterogeneous buildings.

Fig. 3 also illustrates potential interactions among the components, as well as between end-users and the library. The component interactions refer to control inputs (e.g., temperature measurements) and outputs (e.g., temperature setpoints) that can be exchanged between the buildings, DF controls, and analytic applications via the semantics-driven framework. Meanwhile, the user interactions refer to how users can validate the suitability of given semantic models according to the requirements of available controls, as well as browse, select, and export the controls via a GUI. Each component and its interactions are explored in more detail in the following subsections. These will cover the design of their functionalities, present and assess the implemented controls within the library, and evaluate the library's overall benefits through the proposed metrics.

4.1. DFLEXLIBS design

4.1.1. Adherence to design principles

Following the modularity and openness principles outlined in Section 3.1, DFLEXIBS controls are implemented as self-contained functions within a Python package. To support generalizability, they are abstracted from specific building points and designed to work with available data, using flexible semantic queries to retrieve the required points' identifiers for each building. The controls are also adaptable to dynamic operating conditions through logic functions, such as qualification checks that ensure they target zones with sufficient flexibility and suitable conditions. For scalability, the implementation aligns with the recent DF Open Specifications for generalizable, best-practice, rule-based DF controls (Granderson et al., 2025; Granderson & House, 2023). DFLEXIBS also aligns with the discoverability principle through a user interface that enables users to find controls via search, navigation, and validation features, as described in Section 4.1.3.

4.1.2. Adherence to functional and non-functional requirements

To meet the requirements proposed in Section 3.1, all controls are implemented in DFLEXLIBS as implementation codes and have been developed together with supporting software artifacts (semantic queries, configuration files, and interfaces for BOPTTEST and VOLTTRON). These artifacts interface with the semantics-driven framework, illustrated in Fig. 3, to allow the self- and custom-configuration of the controls. This facilitates their portability for testing and performance assessment across the different buildings connected to the BOPTTEST simulation platform (selected for its robust infrastructure supporting controls simulation with Modelica models) and the VOLTTRON EMIS platform (chosen for its open-source nature and use in buildings involved in ongoing projects where DFLEXLIBS has already been tested). Further details on how the semantics-driven framework supports these requirements can be found in previous research (de Andrade et al., 2024).

For performance assessment, DFLEXLIBS includes two main analytic applications: results visualization and benchmarking of KPIs. Results visualization provides predefined plots with representative variables to support the communication and interpretation of insights related to control behavior in each building. Benchmarking involves calculating relevant KPIs to measure and compare the performance of the controls across various buildings. For benchmarking, DFLEXLIBS integrates with the EF KPI Python package, which has been developed within the International Energy Agency (IEA) Annex 81 initiative (Johra et al., 2023).

Several KPIs are available in the EF KPI Python package, and DFLEXLIBS incorporates 12 of them. However, for demonstration, this paper focuses on three KPIs that have proven highly effective in evaluating key areas: demand, cost, and comfort (Blum et al., 2021; Li et al., 2023; Liu et al., 2022; Yin et al., 2023). These KPIs are Demand Decrease Intensity (DDI), Flexibility Savings Index (FSI), and Thermal Discomfort (TD), each of which is described below, and their equations are provided in Appendix A.

- **Demand Decrease Intensity (DDI)** (Liu et al., 2022; Yin et al., 2023) defines the size of a load shed per floor area, averaged across shed events during an evaluation period.
- **Flexibility Savings Index (FSI)** (Li et al., 2023) defines the fraction of saved cost from a flexible control strategy compared with a baseline control strategy.
- **Thermal Discomfort (TD)** (Blum et al., 2021; Li et al., 2023) defines the cumulative deviation of zone temperatures from upper and lower comfort limits, averaged across all zones and over an average day.

4.1.3. DFLEXLIBS GUI prototype

Particularly in compliance with the discoverability principle and categorization requirement, DFLEXLIBS provides a GUI with three main functionalities: search, navigate, and validate (Fig. 4). The *search* function allows users to find controls based on input texts, while the *navigate*

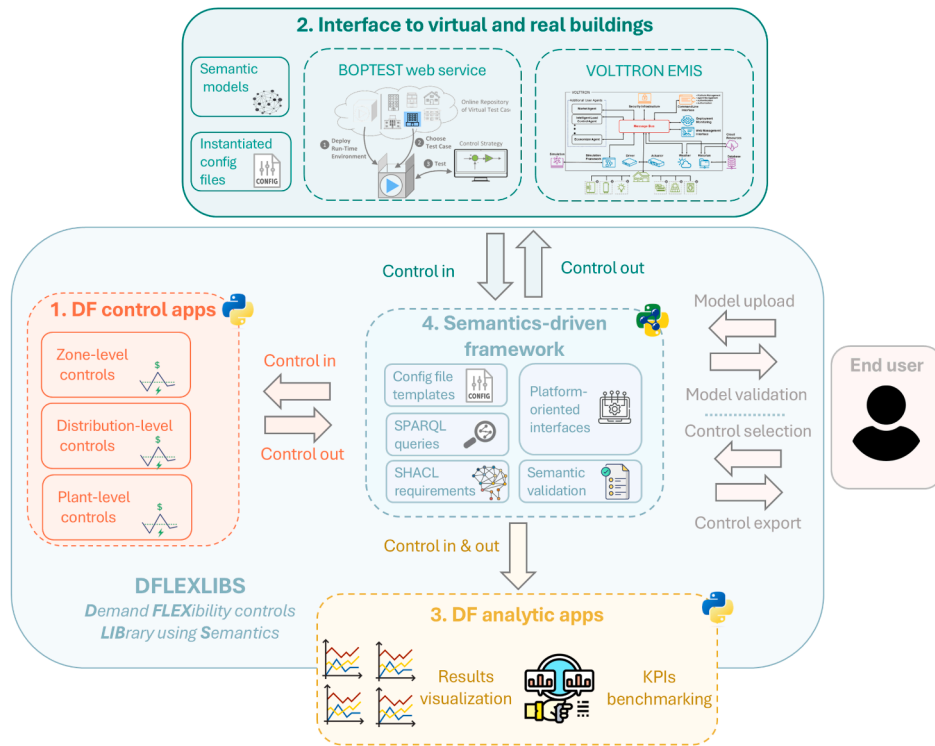


Fig. 3. DFLEXLIBS main components and their software artifacts for implementing, testing and benchmarking DF control applications.

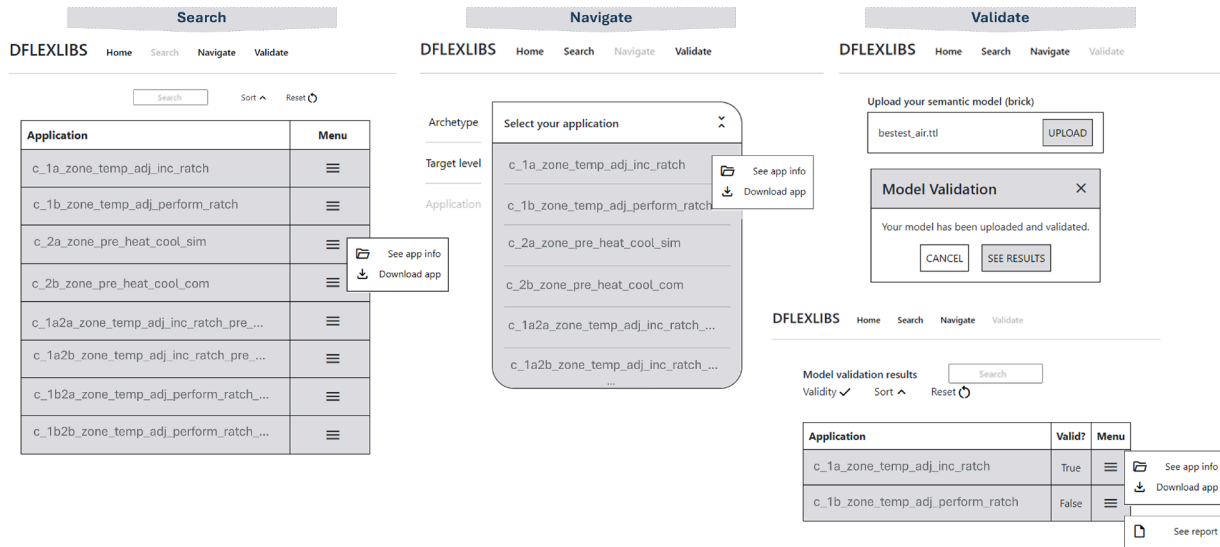


Fig. 4. Snapshots demonstrating the functionalities of the deployed DFLEXLIBS GUI, supporting the discoverability of DF control applications through search, category-based navigation, and semantic validation mechanisms.

function enables browsing controls by categories, such as types of buildings and systems, target levels, and load shapes. The *validate* function helps filter suitable controls for specific buildings by using a semantic validation mechanism. This mechanism is based on BuildingMOTIF (Fierro et al., 2022), as detailed in de Andrade et al. (2024), and compares available data from given buildings’ semantic models with the controls’ data requirements, generating lists of suitable controls and reporting missing metadata for unsuitable ones.

4.2. DFLEXLIBS implementation

The library is implemented in three phases: coding, testing, and assessing. First, controls are designed and coded as scripts applicable to multiple building types. Second, they are tested in BOPTTEST simulations

and real buildings via VOLTTRON EMIS, with semantics-based tools minimizing reconfiguration across sites. Third, control performance is assessed using KPIs (DDI, FSI, TD), leveraging the semantics-driven framework to support portability and helping users understand the factors that influence the DF from each control, the order of magnitude of flexibility in certain buildings and HVAC system types, and the circumstances under which DF performance may be reduced.

4.2.1. DF controls coding

Nine control strategies have been implemented based on the recent work on DF controls open specifications (Granderson et al., 2025; Granderson & House, 2023). These controls, further described in Appendix B, range from zone-level to plant-level strategies. Zone-level strategies (C.1 and C.2) suit all building typologies that count on

writable temperature setpoints at the zone level. They include shedding controls, such as temperature adjustments with incremental temperature and performance target-based ratcheting; shifting controls with both simple and complex modulation for pre-cooling and heating; and composite controls that combine shifting and shedding controls derived from the aforementioned strategies. In contrast, the plant-level strategy (C.3) is tailored specifically for large commercial buildings with built-up chiller systems, focusing on chiller water temperature reset (only for load shedding).

The implementation of the controls followed the main logical functions outlined in Granderson et al. (2025), Granderson and House (2023), including: **zone qualification check** (to qualify zones that are not already challenged to maintain comfort conditions, thereby limiting adverse effects on occupants); **single-step adjustment of zone setpoints** (to enable initial shed or shift for all zones by a given adjustment factor); **ratcheting of select zones** (to allow further load shedding with incremental temperature-based setpoint adjustment, provided that the maximum allowable setpoint has not been reached, or with performance target-based setpoint adjustment to achieve a given shed target, e.g., reduce demand by X kW); **rebound management** (to sequentially release all participating zones in order to avoid a rebound effect); and **target-based demand modulation** (to target zones with the greatest temperature-setpoint deviation, or to adjust the number of zones being to achieve maximum precooling/preheating potential without exceeding a given target, e.g., demand peak). More detailed information about these controls is available in Granderson et al. (2025), Granderson and House (2023).

Fig. 5 illustrates the control flow for some of these functions as implemented in DFLEXLIBS in the context of the zone temperature adjustment strategy with incremental temperature ratcheting (C.1a). As noted, DFLEXLIBS not only incorporates new conditions and commands to manage heating setpoints in addition to cooling, but also introduces two new functions to improve the generalizability potential of the predefined DF controls specification. The first function illustrates how DF events are triggered in response to price signals, and provides guidance on how to effectively manage shed conditions that have already been initiated. The second function is a setback function taking into account zone occupancy and baseline setpoints to conserve energy during shedding events. It allows the override of DF controls in unoccupied zones when prices are high, setting the baseline strategy or enforcing the limit setpoints for load shedding, depending on which are lower for heating and higher for cooling.

In addition to the controls available in the library, DFLEXLIBS also supports adding new controls. The process involves creating new functions if needed, defining the control strategy by combining such functions, and setting up queries and configuration files for parameters not in the semantic models. An interface should then be developed to integrate the control with BOPTTEST, VOLTTRON, or other platforms, handling semantic queries, configuration, and platform-specific data. For BOPTTEST and VOLTTRON, this can largely reuse existing interfaces, with differences mainly in required data types. This is further detailed in our GitHub repository⁴.

4.2.2. DF controls testing

For testing (validating) the generalizability and correct implementation of the controls, we used four simulated (virtual) buildings, realistically modeled in BOPTTEST (Blum et al., 2021), and two real buildings connected to VOLTTRON. Table 2 presents an overview of the location, area, zone, type of building, HVAC system, and the respective tested control applications in each building. More detailed information about the BOPTTEST test cases can be found on the official BOPTTEST page.⁶

For interacting with the simulated buildings and their virtual devices (data ingestion layer in Fig. 2), we leverage the BOPTTEST web service

API⁷ to retrieve measurement data and send new control signals via standard GET and POST operations (abstraction layer in Fig. 2). Similarly, for the real buildings, we use a VOLTTRON agent⁸ to receive measurement data and issue control signals to and from the building points (data ingestion and abstraction layers). Using both the BOPTTEST web service and VOLTTRON agent, we also schedule, process, and log data, as well as access storage databases. These components are then integrated with our controls and software artifacts via our platform-oriented interfaces (operating across the service, integration, and application layers in Fig. 2).

Since all four simulated buildings have heating systems but not all have cooling, and the testing in the real buildings was performed during winter, the DF controls are simulated in BOPTTEST during the predefined two-week period that aligns with the peak heating days of each building, specified in their BOPTTEST testing scenarios. Finally, to trigger the shed events based on grid signals, for the virtual buildings in BOPTTEST we used the dynamic electricity price forecasting, modeled within BOPTTEST based on the buildings' locations, and for the real buildings in VOLTTRON, we used a dynamic electricity price server,⁹ specifically the Winter Multiday Highly Dynamic Price Signal (WinterHDP_MD).¹⁰

Among the implemented strategies, we were unable to test the C.3 application at this time. This is because the real buildings we had access to did not have chiller units, and the BOPTTEST buildings modeling chilled water plants primarily focus on air-side distribution controls and lack overwritable points for adjusting the chilled water temperature. However, two new BOPTTEST buildings currently under development will include chilled water reset set points, allowing us to conduct such tests in the future. Nonetheless, as indicated in Table 2, all C.1 and C.2 control variations were successfully tested across the four BOPTTEST buildings, demonstrating their generalizability and correct implementation. Control C.1a (temperature adjustment with incremental temperature ratcheting) was also tested in the two real buildings.

4.2.3. DF controls assessment

As a demonstration of the library's ability to assess DF controls, this paper will focus on the control strategy C.1a. This is a zone setpoint adjustment control that can be applied across all selected buildings since they all have measurements and setpoints at the zone level. Assessments of the remaining controls solely in BOPTTEST are available in our open-source GitHub repository⁴. The user-defined parameters for the C.1a control (identified in Fig. 5) are configured with the default values from Granderson et al. (2025). Figs. 6 and 7 illustrate the resulting adjustments in the zone temperature setpoints for the control activation, along with the corresponding changes in demand on a representative day for the simulated and real buildings, respectively. Note that while B4 and B5 display setpoint and temperature values for only one of their zones, Fig. 8 presents these values across all zones.

Figs. 6 and 7 show that the six buildings, with different HVAC systems, setpoints, and schedules, produce varied demand profiles and respond differently to shed events despite using the same DF control logic. In general, the commercial buildings (B1, B3, B4, B5, and B6) reach their peak demand in the morning, just before occupancy begins. The simulated commercial buildings (B1, B3, and B5) activate DF shed control during both the morning and afternoon, as these shed events (i.e., high-price periods) align with their occupancy times. In contrast, the real commercial buildings (B5 and B6) only activate the DF control in the

⁷ <https://ibpsa.github.io/project1-boptest/docs-userguide/api.html>

⁸ <https://volttron.readthedocs.io/en/main/agent-framework/agents-overview.html>

⁹ <https://github.com/LBNL-ETA/CalFlexHub>

¹⁰ For demonstration purposes, the WinterHDP_MD signal was shifted by two hours in the morning to align with the occupancy hours of the tested real buildings. This allowed us to have enough shedding time during occupied hours to observe shed control activation with incremental ratcheting, which would not be possible otherwise.

⁶ <https://ibpsa.github.io/project1-boptest/testcases/index.html>

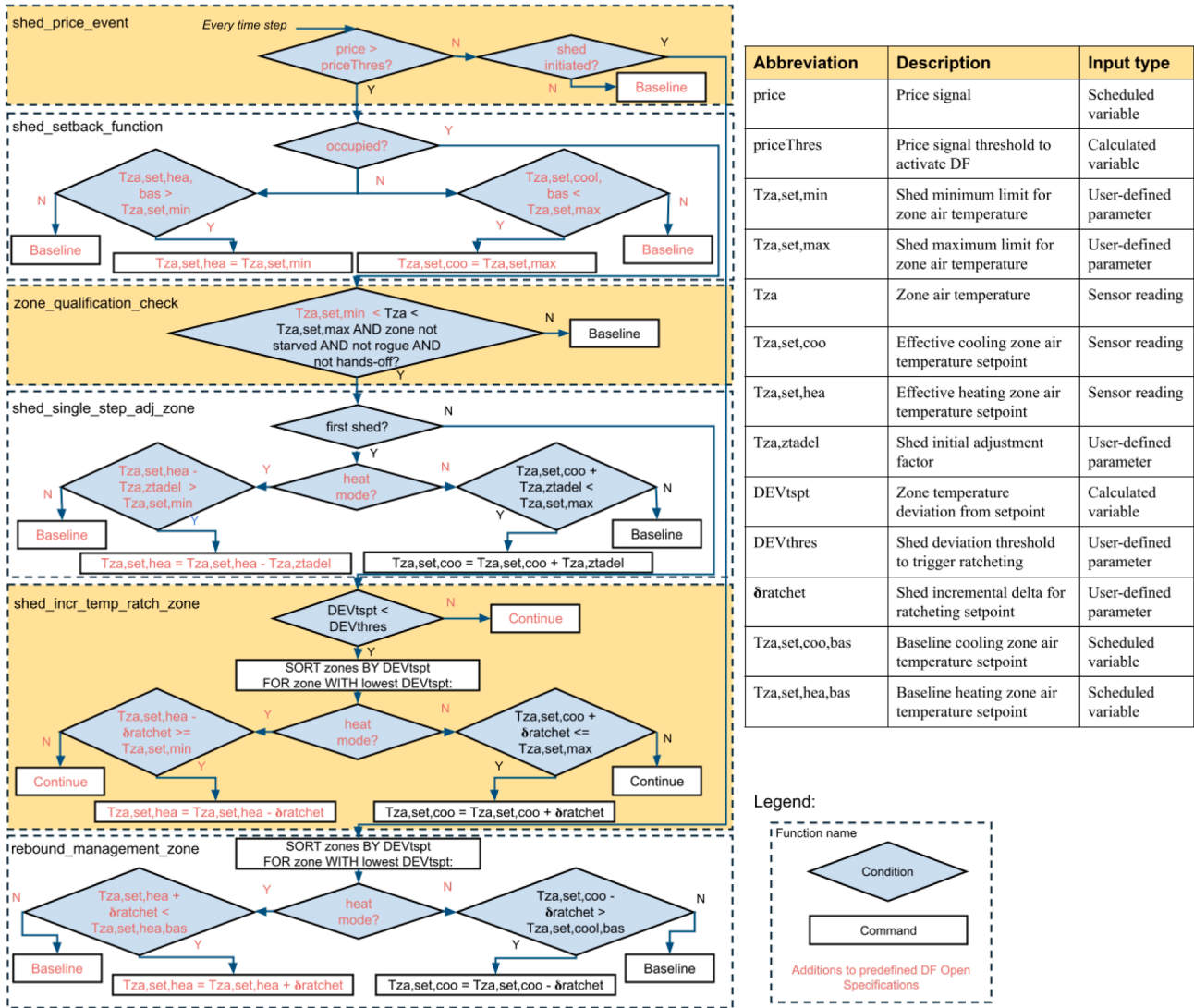


Fig. 5. Control flow diagram for the temperature adjustment strategy with incremental temperature ratcheting (C.1a), including the functions implemented in DFLEXLIBS based on the open specifications in Granderson et al. (2025), Granderson and House (2023).

Table 2

Overview of the tested virtual and real buildings and their respective tested control applications.

| Building (label) | Virtual / Real | Location | Area m ² | Zones | Building type | HVAC system | Tested controls |
|--------------------------------------|----------------|---------------------|---------------------|----------|-------------------------------|---|-----------------|
| Bestest air (B1) | Virtual | Denver, USA | 48 | Single | Small commercial ^a | Idealized four-pipe fan with heating coil served by a gas boiler and cooling coil served by a chiller | All C.1 and C.2 |
| Bestest hydronic heat pump (B2) | Virtual | Brussels, Belgium | 190 | Single | Residential | Air-to-water heat pump | All C.1 and C.2 |
| Single-zone commercial hydronic (B3) | Virtual | Copenhagen, Denmark | 8500 | Single | Large commercial | Air handling unit with heat recovery and radiant heating | All C.1 and C.2 |
| Multi-zone office simple air (B4) | Virtual | Chicago, USA | 1660 | Multiple | Large commercial | Single-duct VAV system with heating coil served by a heat pump and cooling coil served by a chiller | All C.1 and C.2 |
| Site 1 (B5) | Real | California, USA | 630 | Multiple | Medium commercial | Heat-pump rooftop units | C.1a |
| Site 2 (B6) | Real | California, USA | 37.16 | Single | Small commercial | Heat-pump units | C.1a |

^a Although B1 corresponds to a small commercial building due to its size, its HVAC system retains the characteristics of a built-up system.

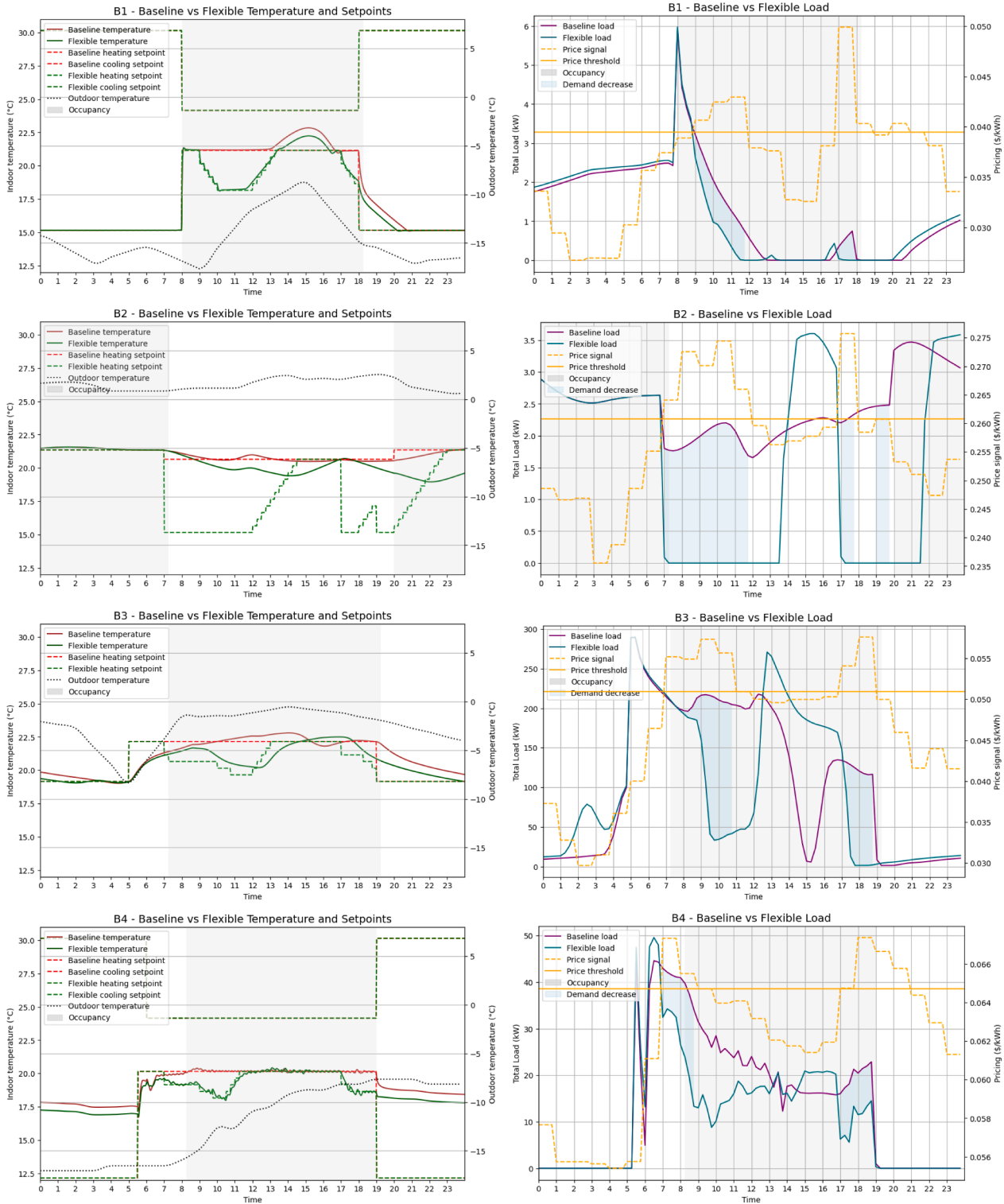


Fig. 6. Adjustment in zone temperature setpoints for activation of the flexible (shed) control C.1a (left) and respective change in the building demand (right) on a representative simulation day in the BOPTEST buildings B1-B4.

morning, since the second shed event occurs in the evening when the buildings are no longer occupied. The residential building B2 peaks in the evening and triggers shed control during unoccupied periods, activating the setback function to reduce setpoints during high-price times.

Despite the different demand profiles, all six buildings consistently respond to the control logic, providing shedding and rebound management with ratcheting. To better understand the ratcheting response in

the tested multi-zone buildings (B4 and B5), Fig. 8 illustrates the changes in temperature and heating temperature setpoint for the five zones of building B4 and the thirteen zones of building B5. Each zone shows the temperature setpoints adjusted incrementally up to their specified limit and according to their current conditions.

Regarding the benchmarking of the six buildings under control C.1a, Table 3 presents results in terms of DDI (demand-related), FSI

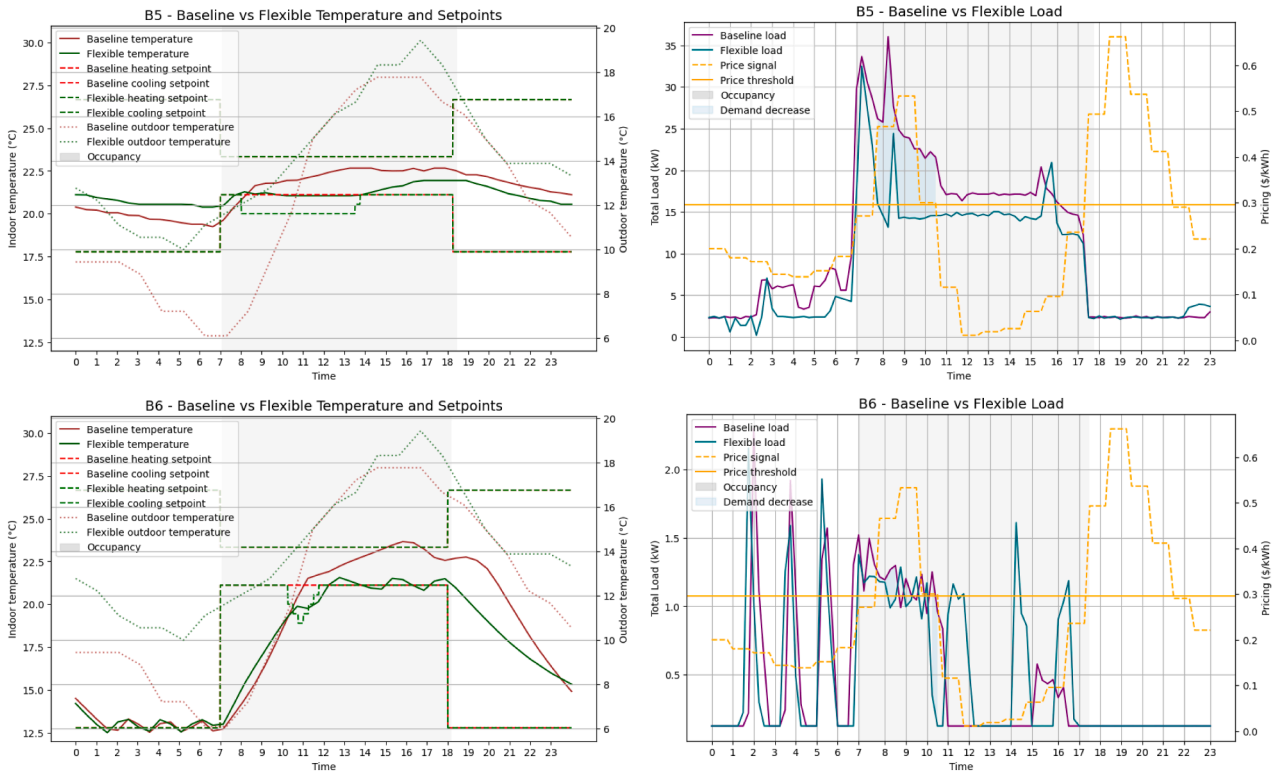


Fig. 7. Adjustment in zone temperature setpoints for activation of the flexible (shed) control C.1a (left) and respective change in the building demand (right) on a baseline and flexible testing day in the real buildings B5 and B6.

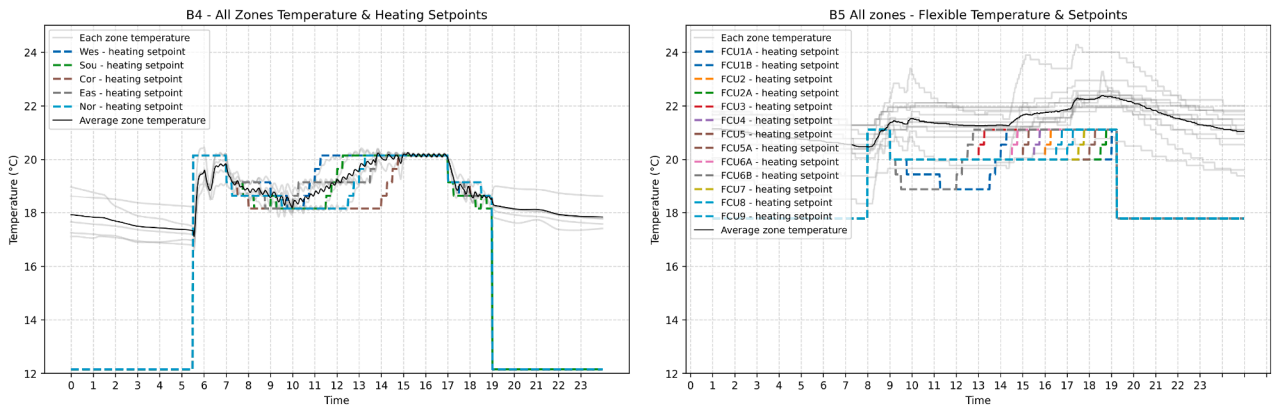


Fig. 8. Illustration of the ratcheting sequence from control strategy C.1a (specifically led by the `shed_incr_temp_ratch_zone` and `rebound_management_zone` functions) across various zones of buildings B4 and B5 on a flexible testing day.

(cost-related), and TD (discomfort-related) performance. The estimated DDI values generally fall within the typical load shed intensity range of 2-10 W/m² reported in the literature (Liu et al., 2022). B3 shows a slightly higher value, around 10.83 W/m², which may be attributed to its longer shedding periods and greater absolute load reduction compared to the other buildings. In contrast, the real building B6, while within the typical range, has a relatively low DDI. This is likely due to a combination of low thermal mass, a cold start, and limited heating capacity. The building loses heat quickly overnight and takes time to warm up, resulting in low indoor temperatures during the morning shed event. This delays DF control qualification for shedding and limits load shed potential (as shown in Fig. 7). However, its shedding potential could be increased by incorporating a pre-heating strategy (an option supported by other control applications in DFLEXLIBS, though not demonstrated in this paper).

In terms of cost savings performance (FSI), B6 also shows the lowest value, for the same reasons observed in its low DDI. Meanwhile, the

Table 3

Benchmarking of the six buildings under control strategy C.1a, evaluated for Demand Decrease Intensity (DDI), Flexibility Savings Index (FSI), and Thermal Discomfort (TD).

| Building | DDI [W/m ²] | FSI [%] | TD [Ch/zone-day] |
|----------|-------------------------|---------|------------------|
| B1 | 6.59 | 6.43 | 10.75 |
| B2 | 7.38 | 31.41 | 7.41 |
| B3 | 10.83 | 6.71 | 2.10 |
| B4 | 5.68 | 21.26 | 11.20 |
| B5 | 7.70 | 28.49 | 1.53 |
| B6 | 2.46 | 4.43 | 18.39 |

other buildings do not follow the same DDI patterns and show more distinct variations in FSI, potentially driven by the following underlying factors.

- B2 shows significant cost savings primarily due to the setback strategy applied during the unoccupied high-price periods. This was effective because B2 had a high setback value during baseline, allowing for better savings during high-price (shed) periods when the building was unoccupied.
- B5 also presents relatively high cost savings. This is likely because the price signal experienced a more pronounced rise from lower to higher values, and B5's load was able to undergo a sharp and responsive reduction when price fluctuations were at their peak, resulting in more concentrated savings.
- B4, despite having a longer duration of high price and extended load reduction, shows relatively smaller savings than B5, potentially because its savings are spread out over time, with less sharp price changes, which leads to a smaller overall savings percentage.
- Finally, B1 and B3 show low overall savings, likely because their loads are primarily thermal rather than electric. As such, since the thermal (gas and district heating) prices for these BOPTTEST buildings remain constant and do not vary like the dynamic electricity prices used as the shedding signal in this test, their cost-saving potential is limited. If the HVAC were a heat pump, using electricity for heating, the potential for cost savings with dynamic prices would have been higher.

In terms of occupant discomfort, B5 exhibits the lowest TD among the six buildings. As illustrated in Fig. 8, the DF control appears to have only a minimal effect on the average indoor temperature. This may be attributed to the building's substantial thermal mass, which helps buffer temperature fluctuations. In addition, the relatively mild outdoor temperatures during the testing period (less severe than those encountered by the other buildings) likely contributed to this outcome as well.

B3 has the second-lowest TD, mainly because its baseline control includes a high heating setpoint, a high setback, and a preheating condition, which keep the building from getting too cold overnight and help compensate for the slow response of its radiant floor system. In contrast, B6 shows the highest TD, likely due to a similar slow-response system, but with no preheating, and a low setback temperature value. This results in a slower temperature rise to the target and causes morning discomfort, even under baseline conditions. Finally, B1, B2, and B4 display similarly high TD values, potentially influenced by their longer shed activation durations and subtle thermal responses (in the case of B1 and B4, as indicated by the rapid indoor temperature changes following the setpoint adjustments), or slow-response system (in the case of B2, with its hydronic system). Although B2 has a high setback temperature in the baseline, which could mitigate this, during flexible control, most unoccupied periods align with high-price (shed) times, triggering the proposed savings, setback strategy.

Although the sample size is limited, the performance assessment offers useful insights into how DF performance varies with building characteristics and operating conditions. It also shows that the three metrics (DDI, FSI, and TD) are not strongly correlated. For instance, large demand reductions did not always yield higher cost savings or greater discomfort, as one might intuitively expect. These results highlight the value of providing diverse KPIs so developers can tailor control choices to their context and goals. If comfort is a priority, developers can shorten shed durations or apply milder setpoint adjustments, especially in low thermal mass buildings. To maximize demand reduction with minimal discomfort, they can pair shedding controls with pre-heating and pre-cooling strategies (which are also available in DFLEXLIBS).

Other stakeholders, such as building portfolio managers, aggregators, and grid operators, can also use results from these performance assessments to drive the selection of target buildings or design price structures in DF programs. For instance, buildings with electric-based HVAC systems have achieved higher cost savings than those using gas or district heating, particularly when dynamic electricity prices are used to trigger shedding events and when gas or district heating prices remain flat. Similarly, greater savings have been observed in scenarios where

electricity prices show a wide spread between off-peak and peak periods. In other words, the more pronounced the price fluctuations, the greater the potential for financial benefits from DF strategies.

4.3. DFLEXLIBS evaluation

To evaluate how DFLEXLIBS reduces control development and deployment effort, this study uses two metrics. Results for the first metric, RR (Eq. (1)), shown in Table 4, indicate rates of about 99% for all controls across BOPTTEST buildings B2-B4, and roughly 48% for the C.1a control across VOLTTRON buildings, using B1 as the reference. This metric considers all LOC used by the controls, including functions and interfaces for semantic queries and configuration files, and the platforms (BOPTTEST and VOLTTRON) setup, data encoding, and decoding.

The high reusability rate among BOPTTEST buildings is due to only two required code changes: one to select the building's identifier, and another to define the logic for identifying the HVAC operation mode (heating or cooling). Since these changes are consistent across all buildings, the variation in RR mostly depends on the total number of lines in each control. As such, controls with more code tend to have higher reusability rates.

In contrast, although the RR between the two VOLTTRON buildings is about 94%, it drops to around 50% between BOPTTEST and VOLTTRON buildings. This mainly reflects changes, additions, or removals of lines in the interface (setup code) due to platform-specific interactions (BOPTTEST simulation vs. VOLTTRON EMIS). Even so, DFLEXLIBS shows clear potential to reduce control development and deployment effort (in terms of LOC) while enabling reusable control code. Note that these evaluated controls are zone-level strategies and can be applied to any building that has zone-level measurements and setpoints. Other strategies, like C3, have more limited applicability and may require building-specific customization.

The calculation of the second metric, FOR (Eq. (2)), is presented in Table 5, highlighting the shared functions among the control applications and their estimated overlap rate. The most commonly shared function, *zone_qualification_check*, is used in 8 out of 9 controls, with the highest overlap rate of 0.89. In contrast, the least shared functions, *chiller_plant_load_check*, *shed_chiller_prioritization*, and *shed_chiller_ratcheting*, each appear in only one control, with an overlap of 0.11.

In general, each control shares at least one function with others, with significant overlaps noted between pairs such as C.1a and C.1b, and C.2a and C.2b. The modular design of these functions also enables seamless integration of the control variations from C.1 (shed) and C.2 (shift) into composite control strategies. This results in the following combinations: C.1a2a, C.1a2b, C.1b2a, and C.1b2b. The control flow for the functions applied to control C.1a is depicted in Fig. 5. More details regarding these and the remaining functions can be found in our open-source GitHub repository⁴.

5. Discussion

This paper presents the successful implementation of a library with DF controls, DFLEXLIBS, developed based on our proposed methodology. A total of nine control sequences were implemented in Python, eight of which were tested across simulated and real buildings, requiring minimal effort. This has been evidenced by the minimal code modifications required between different buildings (particularly those under the same data access platform) and the high degree of functions' overlap among the controls. Such a result supports our original hypothesis, demonstrating the library's potential to reduce controls' development time and deployment (configuration/onboarding) effort.

The following subsections revisit the paper's main contributions: i) design principles and requirements for a controls library; ii) controls portability and performance assessment; and (iii) evaluation of the library's benefits, all in the context of the implemented library, the results obtained, and the broader state-of-the-art.

Table 4

Reusability rates measuring the relative change in LOC between an existing application on a reference building (B1) and the version adapted across the other tested buildings.

| Building | C.1a | C.1b | C.2a | C.2b | C.1a2a | C.1a2b | C.1b2a | C.1b2b |
|----------|-------|-------|-------|-------|--------|--------|--------|--------|
| B1 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 |
| B2 | 99.90 | 99.86 | 99.82 | 99.85 | 99.88 | 99.89 | 99.88 | 99.89 |
| B3 | 99.90 | 99.86 | 99.82 | 99.85 | 99.88 | 99.89 | 99.88 | 99.89 |
| B4 | 99.90 | 99.86 | 99.82 | 99.85 | 99.88 | 99.89 | 99.88 | 99.89 |
| B5 | 48.14 | – | – | – | – | – | – | – |
| B6 | 48.72 | – | – | – | – | – | – | – |

Table 5

Reusability of functions across the control applications implemented within DFLEXLIBS, including overlap rates.

| Function name | C.1a | C.1b | C.2a | C.2b | C.1a2a | C.1a2b | C.1b2a | C.1b2b | C.3 | FOR |
|-----------------------------|------|------|------|------|--------|--------|--------|--------|-----|------|
| shed_price_event | x | x | – | – | x | x | x | x | x | 0.78 |
| shed_setback_function | x | x | – | – | x | x | x | x | – | 0.67 |
| zone_qualification_check | x | x | x | x | x | x | x | x | – | 0.89 |
| shed_single_step_adj_zone | x | x | – | – | x | x | x | x | – | 0.67 |
| shed_incr_temp_ratch_zone | x | – | – | – | x | x | – | – | – | 0.33 |
| shed_perform_target_ratch | – | x | – | – | – | – | x | x | – | 0.33 |
| rebound_management_zone | x | x | – | – | x | x | x | x | – | 0.67 |
| shift_price_occ_event | – | – | x | x | x | x | x | x | – | 0.67 |
| shift_single_step_adj_zone | – | – | x | x | x | x | x | x | – | 0.67 |
| shift_target_demand_mod | – | – | – | x | – | x | – | x | – | 0.33 |
| chiller_plant_load_check | – | – | – | – | – | – | – | – | x | 0.11 |
| shed_chiller_prioritization | – | – | – | – | – | – | – | – | x | 0.11 |
| shed_chiller_ratcheting | – | – | – | – | – | – | – | – | x | 0.11 |

5.1. Design principles and requirements for the library

Recent control libraries have been developed to simplify control design by reducing the need to start from scratch or convert textual specs into code. However, none have explicitly targeted DF or the portability and performance assessment, and many still require niche modeling knowledge (especially for Modelica-based CDL). DFLEXLIBS provides Python-based DF controls that are easily configurable and portable across buildings with semantic models in place. Its modular, building-agnostic control logic allows customization, while integration with BOPTEST enables rapid testing, consistent performance assessment, and greater user confidence.

To create DFLEXLIBS, this paper drew on literature, domain expertise, and practical experience gained throughout the development of this work. We defined functional requirements to guide how controls should be specified, developed, and assessed, including aspects such as packaging and export methods, associated software artifacts, performance assessment strategies, and categorization. For non-functional requirements, we identified how to integrate software artifacts with BAS/EMIS and simulation platforms. Our design principles emphasized modularity, scalability, and generalizability, enabling reusable, best-practice-based functions and easy adaptation to different buildings. We also prioritized openness and discoverability to support an accessible, open-source ecosystem. Although focused on DF controls, the methodology is broadly applicable to creating control libraries for other domains.

5.2. Controls portability and performance assessment

The widespread adoption of emerging control applications depends on two key aspects: cost-effectiveness and proven performance (Blum et al., 2019). The proposed library has shown promise to reduce development cost for DF controls that can be easily assessed in various contexts. Development time is reduced through the use of decoupled interfaces, flexible semantic queries, standard configuration files, and qualification check functions, which enable the integration of control applications across different buildings and data access platforms.

The software interfaces support different methods for reading and writing building data and allow for site-specific logic (e.g., HVAC mode detection, equipment staging, or schedule requirements). Flex-

ible queries support access to varied point names for the same variable (e.g., multiple possible temperature setpoints, such as *Air Temperature Setpoint*, *Heating Zone Air Temperature Setpoint*, *Unoccupied Heating Temperature Setpoint*, or *Occupied Heating Temperature Setpoint*, according to their availability in a building). Standard configuration files capture user-defined DF control settings not found in popular building ontologies (e.g., minimum allowable temperature setpoint during a shed event). Qualification checks help controls to adapt to building-specific characteristics (e.g., units) and real-time operational conditions, such as occupancy, comfort limits, or HVAC status (e.g., starving or rogue conditions).

The performance of the controls has been measured by linking the library's controls to a testing platform, which not only enabled rapid, automated assessment across diverse buildings but also yielded valuable insights (see Section 4.2.3). In general, these insights showed that the performance of DF controls can vary widely between buildings, even when using the same code and configuration. This reinforces the need for a performance assessment before controls deployment in real-world settings, which is currently not common practice in industry.

While testing and assessing controls in a highly accurate simulated model of a target building is ideal, the effort involved can be significant. Instead, testing controls across a range of generic but realistic building models, as proposed in this paper, offers a practical and valuable alternative. It helps debug the controls and provides insights into how different factors and parameters influence their performance. That said, while this work's simulations were based on four prototype virtual buildings, advanced users can model their own buildings in BOPTEST,¹¹ or create additional prototypes, to validate the performance of different control applications.

5.3. Library benefits

In our demonstration using DFLEXLIBS, reference DF controls were coded, tested, and assessed across simulated and real buildings with minimal effort. While the reusability rate reached approximately 50% across platforms (BOPTEST and VOLTTRON), it exceeded 90% within

¹¹ <https://github.com/ibpsa/project1-bopstest/tree/master/docs/tutorials>

the same platform. This indicates that, once an interface and setup code are in place for a given platform, controls can be easily reused across buildings.

Beyond reducing development and deployment efforts, DFLEXLIBS benefits multiple stakeholders. Researchers can advance the state-of-the-art by creating new controls, understanding their performance variability under different scenarios, and promoting their reuse. In turn, industry developers can reuse existing controls or develop new ones based on pre-built functions for deployment in real buildings. This fundamentally reflects the library's modular and portable design.

In addition to testing controls within BOPTTEST, with appropriate interfaces, the controls in DFLEXLIBS could be evaluated through other software-based testing and benchmarking frameworks, such as CityLearn (Nweye et al., 2023), the VOLTTRON-based Building Control Testing (V-BCT) Framework (Huang et al., 2023), or BuildingGym (Dai et al., 2025), provided that they include semantic models of their simulated buildings. These controls could also be deployed in other real buildings via open-source or proprietary BAS, EMIS platforms using a similar approach to the VOLTTRON connection (e.g., creating interfaces to encode and decode data exchanges, whether through APIs or protocols such as BACnet, as well as platform-specific setup).

Looking ahead, DFLEXLIBS could serve as a reference implementation for industry, enabling organizations to create their own versions of the library. Such a library could implement the controls in other programming languages and deploy them as microservices on local servers or cloud platforms (Aksakalli et al., 2021). DFLEXLIBS' underlying framework would help ensure portability across buildings, reducing deployment costs. The proposed GUI shown in Fig. 4 illustrates how developers (or providers, contractors) and aggregators (or other stakeholders) could browse and filter controls using buildings' semantic models, visualize performance under different scenarios and configure controls across building portfolios. This could improve the accessibility and understanding of DF controls and support wider adoption of emerging programs and regulations.

Despite its potential, DFLEXLIBS is a prototype with limitations to address, including: i) more thorough validation of controls with functional tests that could reduce the risk of malfunction and support commissioning procedures; ii) assessment of the library's benefits versus traditional approaches; iii) testing across more buildings and applications for robustness; and iv) sensitivity analysis to understand control performance across various contexts.

6. Conclusion and future work

Cost-effective portability and measurable performance are essential for the scalability of DF controls. This study introduces a method for creating a library of reference, portable DF controls, and presents DFLEXLIBS as a prototype. DFLEXLIBS supports developing and deploying DF controls across buildings with minimal effort. Its main contributions are:

- DFLEXLIBS provides open-source, vendor-independent implementation codes based on best practices, generalizable DF control specifications.
- DFLEXLIBS supports a more automated configuration and the reuse of both DF controls and analytics (KPIs) by leveraging semantic models and appropriate interfaces.
- DFLEXLIBS connects to realistic Modelica buildings through BOPTTEST and to real buildings via VOLTTRON, enabling rapid testing, simulation, and field validation of controls.
- DFLEXLIBS facilitates a quantitative understanding of performance variability of the same code-base DF controls across both simulated and real buildings under various scenarios.

DFLEXLIBS provides a holistic integration of previously fragmented research areas, including controls specification, semantics-driven portability, and performance assessment, offering the scalability the field

needs. It enables researchers and developers to reuse and gather insights from tested applications or create and evaluate their own portable ones across multiple buildings. This can increase confidence in DF controls performance, lower the complexity and cost of making such applications portable, and pave the way for new products and markets.

Future work will enhance DFLEXLIBS by testing DF control for chiller setpoint reset in upcoming BOPTTEST cases, developing new controls, and evaluating existing ones in additional real-world scenarios. We will provide functional tests to verify implementation and gather user feedback to better assess the library's benefits. Finally, we plan to investigate ASHRAE's recently released standards, developing 231-complaint DF control sequences and using 223-complaint semantic models.¹²

CRedit authorship contribution statement

Flavia de Andrade Pereira: Writing – review & editing, Writing – original draft, Visualization, Validation, Software, Methodology, Data curation, Conceptualization; **Marco Pritoni:** Writing – review & editing, Supervision, Software, Methodology, Conceptualization; **Armando Casillas:** Writing – review & editing, Conceptualization; **Jessica Granderson:** Writing – review & editing, Conceptualization; **Lazlo Paul:** Writing – review & editing, Validation, Software, Conceptualization; **Anand Prakash:** Writing – review & editing, Software, Conceptualization; **Conor Shaw:** Writing – review & editing; **Dimitrios Rovas:** Writing – review & editing; **Susana Martin-Toral:** Writing – review & editing; **Donal Finn:** Writing – review & editing; **James O'Donnell:** Writing – review & editing, Supervision, Methodology, Conceptualization.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgement

This work was supported by the CBIM-ETN funded by the European Union's Horizon 2020 research and innovation program under the Marie Skłodowska-Curie grant agreement No 860555; by the Assistant Secretary for Energy Efficiency and Renewable Energy, Building Technologies Office, of the U.S. Department of Energy under Contract No. DE-AC02-05CH11231; by the New York State Energy Research & Development Authority (NYSERDA) through the NextGen HVAC Innovation Challenge program; and by California Energy Commission through grant EPC-19-013 and EPC-24-041. The authors would like to thank Dave Blum at the Lawrence Berkeley National Laboratory for his support and advice, as well as the contributors to the [International Energy Agency \(IEA\) Energy in Buildings and Communities \(EBC\) Annex 81: "Data-Driven Smart Buildings"](https://annex81.iea-ebc.org) (<https://annex81.iea-ebc.org>) for fostering an excellent research network, particularly Subtask C3, which supported the development of the EF KPI Python package.

Appendix A. Energy flexibility kpis equations

Demand Decrease Intensity (DDI) is expressed as W/m^2 and calculated as:

$$DDI = \frac{1}{A} \cdot \frac{1}{N} \sum_{t \in [t_s, t_f]} (D_{baseline}(t) - D_{flexible}(t)) \quad (A.1)$$

where $D_{baseline}$ is the demand during the baseline control, $D_{flexible}$ is the demand during the flexible control, N is the number of shed events,

¹² <https://www.ashrae.org/technical-resources/standards-and-guidelines/titles-purposes-and-scopes>.

A is the building floor area, t_s and t_f are the start time and the end time of the evaluation period.

Flexibility Savings Index (FSI) is expressed as a percentage (%) and calculated as:

$$FSI = 1 - \frac{C_{flexible}}{C_{baseline}} \tag{A.2}$$

where $C_{flexible}$ is the cost of operation during the flexible control and $C_{baseline}$ is the cost of operation during the baseline control.

Thermal Discomfort (TD) is expressed as °C·h / zone·day and calculated as:

$$TD = \frac{1}{N} \cdot \frac{1}{D} \sum_{z=1}^N \int_{t_s}^{t_f} (\max(T_z(t) - T_{z,cool}(t), 0) + \max(T_{z,heat}(t) - T_z(t), 0)) dt \tag{A.3}$$

where $T_z(t)$ is the zone temperature of zone z , $T_{z,cool}(t)$ and $T_{z,heat}(t)$ are the cooling and heating set points of zone z , N is the number of zones, D is the number of days from the evaluated period, and t_s and t_f are the start time and the end time of the evaluation period.

Appendix B. DFLEXLIBS controls

Table B.1 describes the DF control strategies implemented in DFLEXLIBS, based on the recent work on DF controls open specifications (Granderson et al., 2025; Granderson & House, 2023).

Table B.1
DF control strategies in DFLEXLIBS.

| Target level | Control strategy group | Control strategy | Building archetype | Description |
|-------------------------------|--|---|---|--|
| Zone-level target | C1_Zone temperature adjustment | C1a_Incremental temperature ratcheting ^a | Residential buildings (e.g., split systems); Small and medium commercial buildings (e.g., package units); Large commercial buildings (e.g., build-up systems) | Strategy enables load shedding by adjusting zone setpoints within comfort limits, raising cooling setpoints (or lowering heating setpoints) by a defined single-step adjustment factor with incremental, temperature-based setpoint changes. |
| Zone-level target | C1_Zone temperature adjustment | C1b_Performance target-based ratcheting ^b | Residential buildings (e.g., split systems); Small and medium commercial buildings (e.g., package units); Large commercial buildings (e.g., build-up systems) | Strategy enables load shedding by adjusting zone setpoints within comfort limits, raising cooling setpoints (or lowering heating setpoints) by a defined single-step adjustment factor while following a performance-target approach to meet a specified shed target (e.g., reduce demand by X kW). |
| Zone-level target | C2_Zone temperature adjustment for precooling/heating | C2a_Simple target-based demand modulation ^b | Residential buildings (e.g., split systems); Small and medium commercial buildings (e.g., package units); Large commercial buildings (e.g., build-up systems) | Strategy adjusts zone setpoints within comfort limits prior to a peak event, lowering cooling setpoints (or raising heating setpoints) to precool (or preheat) the building mass. It prioritizes zones with the greatest deviation from setpoint. |
| Zone-level target | C2_Zone temperature adjustment for precooling/heating | C2b_Complex target-based demand modulation ^b | Residential buildings (e.g., split systems); Small and medium commercial buildings (e.g., package units); Large commercial buildings (e.g., build-up systems) | Strategy adjusts zone setpoints within comfort limits prior to a peak event, lowering cooling setpoints (or raising heating setpoints) to precool (or preheat) the building mass. It dynamically selects the number of participating zones to maximize precooling/preheating potential without exceeding a specified target (e.g., demand peak limit). |
| Zone-level target (composite) | C1_Zone temperature adjustment & C2_Zone temperature adjustment for precooling/heating | C1a2a_Incremental temperature ratcheting & Simple target-based demand modulation ^b | Residential buildings (e.g., split systems); Small and medium commercial buildings (e.g., package units); Large commercial buildings (e.g., build-up systems) | Combine C1a and C2a. |
| Zone-level target (composite) | C1_Zone temperature adjustment & C2_Zone temperature adjustment for precooling/heating | C1a2b_Incremental temperature ratcheting & Complex target-based demand modulation ^b | Residential buildings (e.g., split systems); Small and medium commercial buildings (e.g., package units); Large commercial buildings (e.g., build-up systems) | Combine C1a and C2b. |
| Zone-level target (composite) | C1_Zone temperature adjustment & C2_Zone temperature adjustment for precooling/heating | C1b2a_Performance target-based ratcheting & Simple target-based demand modulation ^b | Residential buildings (e.g., split systems); Small and medium commercial buildings (e.g., package units); Large commercial buildings (e.g., build-up systems) | Combine C1b and C2a. |
| Zone-level target (composite) | C1_Zone temperature adjustment & C2_Zone temperature adjustment for precooling/heating | C1b2b_Performance target-based ratcheting & Complex target-based demand modulation ^b | Residential buildings (e.g., split systems); Small and medium commercial buildings (e.g., package units); Large commercial buildings (e.g., build-up systems) | Combine C1b and C2b. |
| Plant-level target | C3_Chiller water temperature setpoint reset | C3_Chiller water temperature setpoint reset | Large commercial buildings (e.g., build-up systems) | Strategy enables load shedding by increasing the chilled water temperature setpoint of chiller plants. |

^a strategy tested in simulation and field

^b strategy tested in simulation

References

- ASHRAE Guideline 36-2018 - High-Performance Sequences of Operation for HVAC Systems. (2021).
- Aksakalli, I. K., Celik, T., Can, A. B., & Tekinerdogan, B. (2021). Deployment and communication patterns in microservice architectures: A systematic literature review. *Journal of Systems and Software*, 180, 111014. <https://doi.org/10.1016/j.jss.2021.111014>
- Building automation and control systems (BACS) – part 3: Functions, (2005). (ISO 16484-3:2005). <https://www.iso.org/standard/31980.html>.
- Balaji, B., Bhattacharya, A., Fierro, G., Gao, J., Gluck, J., Hong, D., Johansen, A., Koh, J., Ploennigs, J., Agarwal, Y., Bergés, M., Culler, D., Gupta, R. K., Kjærsgaard, M. B., Srivastava, M., & Whitehouse, K., et al. (2018). Brick : Metadata schema for portable smart building applications. *Applied Energy*, 226, 1273–1292. <https://doi.org/10.1016/j.apenergy.2018.02.091>
- Bartusiak, R. D., Bitar, S., DeBari, D. L., Houk, B. G., Stevens, D., Fitzpatrick, B., & Sloan, P. (2022). Open process automation: A standards-based, open, secure, interoperable process control architecture. *Control Engineering Practice* 121, 105034. <https://doi.org/10.1016/j.conengprac.2021.105034>
- Bennani, I. L., Prakash, A. K., Zafiriz, M., Paul, L., Roa, C. D., Raftery, P., Pritoni, M., & Fierro, G., et al. (2021). Query relaxation for portable brick-based applications. In *Proceedings of the 8th ACM international conference on systems for energy-efficient buildings, cities, and transportation* (pp. 150–159). Coimbra Portugal: ACM. <https://doi.org/10.1145/3486611.3486671>
- Bergmann, H., Mosiman, C., Saha, A., Haile, S., Livingood, W., Bushby, S., Fierro, G., Bender, J., Poplawski, M., Granderson, J., & Pritoni, M., et al. (2020). Semantic interoperability to enable smart, grid-interactive efficient buildings, (p. 18). <https://doi.org/10.20357/B7S304>
- Blum, D., Arroyo, J., Huang, S., Drgoňa, J., Jorissen, F., Walnum, H. T., Chen, Y., Benne, K., Vrabie, D., Wetter, M., & Helsen, L. (2021). Building optimization testing framework (BOPTTEST) for simulation-based benchmarking of control strategies in buildings. *Journal of Building Performance Simulation*, 14(5), 586–610. <https://doi.org/10.1080/19401493.2021.1986574>
- Blum, D., Jorissen, F., Huang, S., Chen, Y., Arroyo, J., Benne, K., Li, Y., Gavan, V., Rivalin, L., Helsen, L., Vrabie, D., Wetter, M., & Sofos, M., et al. (2019). Prototyping the BOPTTEST framework for simulation-based testing of advanced control strategies in buildings. (pp. 2737–2744). Rome, Italy. <https://doi.org/10.26868/25222708.2019.211276>
- Chamari, L., Petrova, E., & Pauwels, P. (2023a). An end-to-end implementation of a service-oriented architecture for data-driven smart buildings. *IEEE Access*, 11, 117261–117281. <https://doi.org/10.1109/ACCESS.2023.3325767>
- Chamari, L., Petrova, E., & Pauwels, P., et al. (2023b). Extensible real-time data acquisition and management for IoT enabled smart buildings. *European Council on Computing in Construction* (vol. 4). Computing in Construction. <https://doi.org/10.35490/EC3.2023.300>
- Chamari, L., Walker, S., Petrova, E., & Pauwels, P. (2025). Towards portable model predictive control-based applications for demand side management in buildings. *Energy and Buildings*, 347, 116257. <https://doi.org/10.1016/j.enbuild.2025.116257>
- Chiosa, R., Piscitelli, M., Pritoni, M., & Capozzoli, A. (2024). A portable application framework for energy management and information systems (EMIS) solutions using brick semantic schema. *Energy and Buildings*, 323, 114802. <https://doi.org/10.1016/j.enbuild.2024.114802>
- Dai, X., Chen, R., Guan, S. et al. (2025). Buildinggym: An open-source toolbox for AI-based building energy management using reinforcement learning. *Building Simulation*, 18, 1909–1927. <https://doi.org/10.1007/s12273-025-1306-y>
- de Andrade, F., Katsigarakis, K., Rovas, D., Pritoni, M., Shaw, C., Paul, L., Prakash, A., Martin-Toral, S., Finn, D., & O'Donnell, J. (2025). A semantics-driven framework to enable demand flexibility control applications in real buildings. *Advanced Engineering Informatics*, 64, 103049. <https://doi.org/10.1016/j.aei.2024.103049>
- de Andrade, F., Paul, L., Pritoni, M., Casillas, A., Prakash, A., Huang, W., Shaw, C., Martin-Toral, S., Finn, D., & Donnell, J. O. (2024). Enabling portable demand flexibility control applications in virtual and real buildings. *Journal of Building Engineering*, (p. 108645). <https://doi.org/10.1016/j.jobe.2024.108645>
- de Andrade, F., Katsigarakis, K., Rovas, D., Korbakis, G., Kostis, N., Serepas, F., & Palencia, S. M. (2024). Deliverable D3.1 - Technical requirements and architecture design. Technical Report University College London.
- Duarte Roa, C., Raftery, P., Prakash, A., & Peffer, T. (2023). Field Demonstration of the Brick Ontology to Scale up the Deployment of ASHRAE Guideline 36 Control Sequences. Technical Report Lawrence Berkeley National Laboratory. <https://escholarship.org/uc/item/53c0t2b1>.
- Faulkner, C. A., Lutes, R., Huang, S., Zuo, W., & Vrabie, D. (2023). Simulation-based assessment of ASHRAE guideline 36, considering energy performance, indoor air quality, and control stability. *Building and Environment*, 240, 110371. <https://doi.org/10.1016/j.buildenv.2023.110371>
- Fierro, G., Pritoni, M., Abdelbaky, M., Lengyel, D., Leyden, J., Prakash, A., Gupta, P., Raftery, P., Peffer, T., Thomson, G., & Culler, D. E., et al. (2019). Mortar: An open testbed for portable building analytics. *ACM Transactions on Sensor Networks*, 16(1). <https://doi.org/10.1145/3366375>
- Fierro, G., Saha, A., Shapinsky, T., Steen, M., & Eslinger, H., et al. (2022). Application-driven creation of building metadata models with semantic sufficiency. In *Proceedings of the 9th ACM international conference on systems for energy-efficient buildings, cities, and transportation BuildSys '22* (pp. 228–237). New York, NY, USA: Association for Computing Machinery. <https://doi.org/10.1145/3563357.3564083>
- Fu, Y., O'Neill, Z., Wen, J., Pertzborn, A., & Bushby, S. T. (2022). Utilizing commercial heating, ventilating, and air conditioning systems to provide grid services: A review. *Applied Energy*, 307. <https://doi.org/10.1016/j.apenergy.2021.118133>
- Granderson, J., Casillas, A., & Pritoni, M. (2025). An open control sequence specification to scale building demand flexibility via analytics software. *Energy and Buildings*. <https://doi.org/10.1016/j.enbuild.2025.115616>
- Granderson, J., & House, J. (2023). Implementation guidance for fault-free optimal control. Technical Report Berkeley Lab. <https://transformingbuildingcontrols.lbl.gov/>.
- He, F., Deng, Y., Xu, Y., Xu, C., Hong, D., & Wang, D., et al. (2021). Energon: A data acquisition system for portable building analytics. In *Proceedings of the twelfth ACM international conference on future energy systems* (pp. 15–26). Virtual Event Italy: ACM. <https://doi.org/10.1145/3447555.3464850>
- Holmström, J., Ketokivi, M., & Hameri, A.-P. (2009). Bridging practice and theory: A design science approach. *Decision Sciences*, 40(1), 65–87. <https://doi.org/10.1111/j.1540-5915.2008.00221.x>
- Huang, S., Lutes, R., Faulkner, C. A., Vrabie, D. L., Katipamula, S., & Zuo, W. (2023). An open-source framework for simulation-based testing of buildings control strategies. *Journal of Building Performance Simulation*, 16(6), 631–643. <https://doi.org/10.1080/19401493.2023.2191220>
- International Energy Agency (2023). Demand response. (accessed 13 April 2024) <https://www.iea.org/energy-system/energy-efficiency-and-demand/demand-response>.
- International Energy Agency (2024). World energy investment 2024. (accessed 25 June 2024) <https://www.iea.org/reports/world-energy-investment-2024>.
- Johra, H., Li, H., de Andrade, F., Nweye, K., Chamari, L., & Nagy, Z. (2023). Iea ebc annex 81 data-driven smart buildings: Building-to-grid applications. In *Proceedings of building simulation 2023: 18th conference of IBPSA* (pp. 2530–2537). Shanghai, China: IBPSA (vol. 18). Building Simulation. <https://doi.org/10.26868/25222708.2023.1305>
- Katipamula, S., Haack, J., Hernandez, G., Akyol, B., & Hagerman, J. (2016). Volttron: An open-source software platform of the future. *IEEE Electrification Magazine*, 4(4), 15–22. <https://doi.org/10.1109/MELE.2016.2614178>
- Kucera, A., & Pitner, T. (2018). Semantic BMS: Allowing usage of building automation data in facility benchmarking. *Advanced Engineering Informatics*, 35, 69–84. <https://doi.org/10.1016/j.aei.2018.01.002>
- Li, H., & Hong, T. (2022). A semantic ontology for representing and quantifying energy flexibility of buildings. *Advances in Applied Energy*, 8, 100113. <https://doi.org/10.1016/j.aaden.2022.100113>
- Li, H., Johra, H., de Andrade Pereira, F., Hong, T., Le Dréau, J., Maturo, A., Wei, M., Liu, Y., Saberi-Derakhtenjani, A., Nagy, Z., Marszal-Pomianowska, A., Finn, D., Miyata, S., Kaspar, K., Nweye, K., O'Neill, Z., Pallonetto, F., & Dong, B., et al. (2023). Data-driven key performance indicators and datasets for building energy flexibility: A review and perspectives. *Applied Energy*, 343. <https://doi.org/10.1016/j.apenergy.2023.121217>
- Li, H., Wang, Z., Hong, T., & Piette, M. A. (2021). Energy flexibility of residential buildings: A systematic review of characterization and quantification methods and applications. *Advances in Applied Energy*, 3, 100054. <https://doi.org/10.1016/j.aaden.2021.100054>
- Liu, J., Yin, R., Yu, L., Piette, M. A., Pritoni, M., Casillas, A., Xie, J., Hong, T., Neukomm, M., & Schwartz, P. (2022). Defining and applying an electricity demand flexibility benchmarking metrics framework for grid-interactive efficient commercial buildings. *Advances in Applied Energy*, 8, 100107. <https://doi.org/10.1016/j.aaden.2022.100107>
- Liu, J., Yu, L., Yin, R., Piette, M., Pritoni, M., Casillas, A., & Schwartz, P. (2023). Factors influencing building demand flexibility. Technical Report Lawrence Berkeley National Laboratory. <https://doi.org/10.20357/B71882>
- Luo, N., Langevin, J., Chandra-Putra, H., & Lee, S. H. (2022). Quantifying the effect of multiple load flexibility strategies on commercial building electricity demand and services via surrogate modeling. *Applied Energy*, 309, 118372. <https://doi.org/10.1016/j.apenergy.2021.118372>
- Maier, L., Jansen, D., Wüllhorst, F., Kremer, M., Kümpel, A., Blacha, T., & Müller, D. (2023). Aixlib: An open-source modella library for compound building energy systems from component to district level with automated quality management. *Journal of Building Performance Simulation*, 0(0), 1–24. <https://doi.org/10.1080/19401493.2023.2250521>
- Martin, A. J., Banyard, C. P., Research, B. S., & Association, I. (1998). Library of System Control Strategies. Application Guides Series. Building Services Research & Information Association. https://books.google.lk/books?id=e_53AAAACAAJ.
- Mavropapnidis, D., Fierro, G., Husmann, M., Korolija, I., & Rovas, D. (2023a). SEEQ: A programming model for portable data-driven building applications. BuildSys '23: The 10th ACM International Conference on Systems for Energy-Efficient Buildings, Cities, and Transportation. <https://doi.org/10.1145/3600100.3623744>
- Mavropapnidis, D., Fierro, G., Korolija, I., & Rovas, D., et al. (2023b). A programming model for portable fault detection and diagnosis. In *Proceedings of the 14th ACM international conference on future energy systems e-Energy '23* (pp. 127–131). New York, NY, USA: Association for Computing Machinery. <https://doi.org/10.1145/3575813.3595190>
- Motegi, N., Piette, M. A., Watson, D., Kilicote, S., & Xu, P. (2007). Introduction to commercial building control strategies and techniques for demand response, . (p. 76). <https://eta-publications.lbl.gov/sites/default/files/59975.pdf>.
- Nweye, K., Kaspar, K., Buscemi, G., Pinto, G., Li, H., Hong, T., Ouf, M., Capozzoli, A., & Nagy, Z. (2023). Citylearn v2: An openAI gym environment for demand response control benchmarking in grid-interactive communities. In *Proceedings of the 10th ACM international conference on systems for energy-efficient buildings, cities, and transportation BuildSys '23* (p. 274–275). New York, NY, USA: Association for Computing Machinery. <https://doi.org/10.1145/3600100.3626257>
- Paul, L., De Andrade Pereira, F., Prakash, A., Ham, S. w., Pritoni, M., Brown, R., & Feng, J. D. (2024). Open building operating system: A grid responsive semantics-driven control platform for buildings. *Science and Technology for the Built Environment*, (p. 294–311). <https://doi.org/10.1080/23744731.2024.2444819>

- Pauwels, P., & Fierro, G. (2022). A reference architecture for data-driven smart buildings using brick and LBD ontologies. *CLIMA 2022 conference*. <https://doi.org/10.34641/clima.2022.425>
- Rogers, Y., Sharp, H., & Preece, J. (2023). *Interaction Design: Beyond Human-Computer Interaction*. (6th ed.). John Wiley & Sons.
- Roth, A., Wetter, M., Benne, K., Blum, D., Chen, Y., Fierro, G., Pritoni, M., Saha, A., & Vrabie, D. (2022). Towards digital and performance-based supervisory HVAC control delivery. ACEEE Summer Study on Energy Efficiency in Buildings. <https://doi.org/10.20357/B70G62>
- Santos, A. Q., Energy in buildings and communities programme, annex energy flexible buildings, teknologisk institut, technology collaboration programme (2019). *Control strategies and algorithms for obtaining energy flexibility in buildings*. Taastруп: Danish Technological Institute. <https://www.annex67.org/media/1898/control-strategies.pdf>.
- Schneider, G. F., Peßler, G., & Steiger, S. . Modelling and simulation of standardised control functions from building automation. In *Proceedings of 12th international modelica conference*. Prague, Czech Republic. <https://doi.org/10.3384/ecp17132209>
- Schraven, M., Kumpel, A., Baranski, M., Mersch, M., Storek, T., Bode, G., Nurenb-erg, M., Vering, C., & Muller, D. (2019). AixOCAT—introducing a library for building automation and control systems. *The REHVA European HVAC Journal*, 56(4), 30–33.
- Tomasevic, N. M., Batic, M. C., Blanes, L. M., Keane, M. M., & Vranes, S. (2015). Ontology-based facility data model for energy management. *Advanced Engineering Informatics*, 29(4), 971–984. <https://doi.org/10.1016/j.aei.2015.09.003>
- Tung, Y.-C., Bui, D., & Shin, K. G. (2018). Cross-platform support for rapid development of mobile acoustic sensing applications. In *Proceedings of the 16th annual international conference on mobile systems, applications, and services MobiSys '18* (p. 455-467). New York, NY, USA: Association for Computing Machinery. <https://doi.org/10.1145/3210240.3210312>
- U.S. Energy Information Administration (2023). Solar and wind power curtailments rising in california. (accessed 06 June 2024) <https://www.eia.gov/todayinenergy/detail.php?id=60822>.
- Umran Alrubae, A., Cetinkaya, D., Liebchen, G., & Dogan, H. (2020). A process model for component-based model-driven software development. *Information*, 11(6). <https://www.mdpi.com/2078-2489/11/6/302>. <https://doi.org/10.3390/info11060302>
- Wan, L., Rossa, F., Welfonder, T., Petrova, E., & Pauwels, P. (2025). Enabling scalable model predictive control design for building HVAC systems using semantic data modelling. *Automation in Construction*, 170, 105929. <https://doi.org/10.1016/j.autcon.2024.105929>
- Waterworth, D., Sethuvenkatraman, S., & Sheng, Q. Z. (2021). Advancing smart building readiness: Automated metadata extraction using neural language processing methods. *Advances in Applied Energy*, 3, 100041. <https://doi.org/10.1016/j.adapen.2021.100041>
- Wetter, M., Benne, K., Gautier, A., Nouidui, T., Ramle, A., & Roth, A. (2020). Lifting the garage door on Spawn, and open-source BEM controls engine. *Building Performance Modeling Conference and SimBuild co-organized by ASHRAE and IBPSA-USA*. <https://escholarship.org/uc/item/9c28b4qp>.
- Wetter, M., Ehrlich, P., Gautier, A., Grahovac, M., Haves, P., Hu, J., Prakash, A., Robin, D., & Zhang, K., et al. (2022). OpenBuildingControl: Digitizing the control delivery from building energy modeling to specification, implementation and formal verification. *Energy*, 238. <https://doi.org/10.1016/j.energy.2021.121501>
- Wetter, M., Grahovac, M., & Hu, J. (2018). Control description language. In *Proceedings of the 1st american modelica conference*. Cambridge, MA, USA. <https://doi.org/10.3384/ECP1815417>
- Wetter, M., Hu, J., Prakash, A., Ehrlich, P., Fierro, G., Grahovac, M., Pritoni, M., Rivalin, L., & Robin, D., et al. (2021). Modelica-json: Transforming energy models to digitize the control delivery process. <https://doi.org/10.26868/25222708.2021.30141>
- Wetter, M., Zuo, W., Nouidui, T. S., & Pang, X. (2014). Modelica buildings library. *Journal of Building Performance Simulation*, 7(4), 253–270. <https://doi.org/10.1080/19401493.2013.765506>
- Willhorst, F., Maier, L., Jansen, D., Kühn, L., Hering, D., & Mueller, D. (2023). Besmod - a modelica library providing building energy system modules. (pp. 9–18). <https://doi.org/10.3384/ECP211869>
- Xu, K., Chen, Z., Xiao, F., Zhang, J., Zhang, H., & Ma, T. (2024). Semantic model-based large-scale deployment of AI-driven building management applications. *Automation in Construction*, 165, 105579. <https://doi.org/10.1016/j.autcon.2024.105579>
- Yin, R., Kara, E. C., Li, Y., DeForest, N., Wang, K., Yong, T., & Stadler, M. (2016). Quantifying flexibility of commercial and residential loads for demand response using setpoint changes. *Applied Energy*, 177, 149–164. <https://doi.org/10.1016/j.apenergy.2016.05.090>
- Yin, R., Liu, J., Piette, M. A., Xie, J., Pritoni, M., Casillas, A., Yu, L., & Schwartz, P. (2023). Comparing simulated demand flexibility against actual performance in commercial office buildings. *Building and Environment*, 243, 110663. <https://doi.org/10.1016/j.buildenv.2023.110663>
- Zhang, K., Blum, D., Cheng, H., Paliaga, G., Wetter, M., & Granderson, J. (2022). Estimating ASHRAE guideline 36 energy savings for multi-zone variable air volume systems using spawn of energyplus. *Journal of Building Performance Simulation*, 15(2), 215–236. <https://doi.org/10.1080/19401493.2021.2021286>