Artifact: Co-zyBench: a thermal comfort provision benchmark for smart buildings

Jun Ma, Dimitrije Panic, Roberto Yus, Georgios Bouloukakis

Artifact: Co-zyBench: A Thermal Comfort Provision Benchmark for Smart Buildings

Jun Ma†, Dimitrije Panic†, Roberto Yus*, Georgios Bouloukakis†
{jun_ma, dimitri.panic, georgios.bouloukakis}@telecom-sudparis.eu, ryus@umbc.edu
†Télécom SudParis, Institut Polytechnique de Paris, France
*Dept. of Computer Science and Electrical Engineering, University of Maryland, Baltimore County, USA

I. Introduction

This paper presents a documentation and testing artifact for Co-zyBench [1], a benchmarking platform to evaluate and optimize thermal comfort provision systems in smart buildings. Co-zyBench is based on a co-simulation middleware that mediates between a Digital Twin (DT) of the smart building and HVAC system and a DT of the occupants of the building that simulates their dynamic thermal preference in different spaces. The DTs that support Co-zyBench are generated based on information about the space in which the thermal comfort system has to be evaluated. Co-zyBench includes a set of metrics that consider energy consumption, thermal comfort, and equality towards the occupants. Also, the benchmark includes a set of DTs based on standard buildings and HVAC systems and occupants (with different thermal preference distributions to represent diversity). Co-zyBench is based on the following components (see Figure 1):

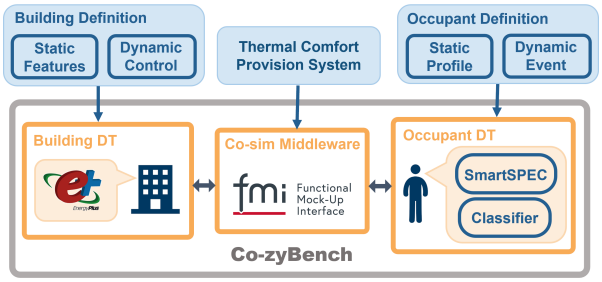

Fig. 1: Overview of Co-zyBench.

a) Building DT: simulates the building and its HVAC system using EnergyPlus[1]. The DT is built by two types of inputs: 1) *Static Features* including descriptions of the building's floorplan, construction materials, and the HVAC system; 2) *Dynamic Control* which covers operational parameters such as control frequencies and specific control strategies, e.g., how to translate group thermal comfort levels to HVAC setpoints.

b) Occupant DT: simulates occupant movement and thermal sensation using SmartSPEC [2] and a KNN Classifier, respectively. These tools utilize two types of inputs: 1) *Static Profile* defining occupant thermal preference profiles; 2) *Dynamic Event* which captures the interactions among occupants and the events that occur inside the building.

[1]https://energyplus.net/

c) Co-simulation Middleware: is implemented using the Functional Mock-Up Interface (FMI)[2] and is key in enabling smooth integration and interaction among the DTs.

II. Using Co-zyBench

As depicted in Figure 2, Co-zyBench can be used as follows: First, the user selects the thermal comfort provision system(s) intended for evaluation (**Step 1**). Then they select a benchmark scenario which can be either: their own DTs (**Step 2.a**) based on their specific buildings; or predefined DTs (**Step 2.b**) including reference scenarios such as offices, hospitals, and hotels. Finally, the user runs the selected benchmark (**Step 3**) and obtains the evaluation results (**Step 4**).

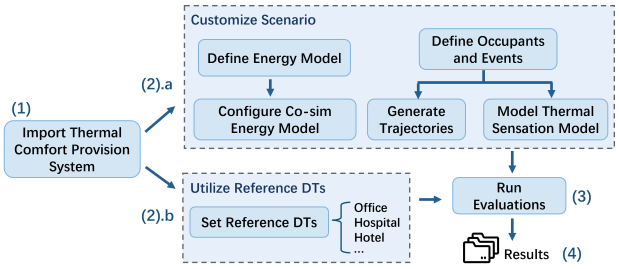

Fig. 2: Co-zyBench workflow.

A. System Requirements and Installation

Co-zyBench, available on GitHub [3], has been tested on Windows 10 and 11 (for other OS we recommend using a Windows virtual machine to ensure compatibility). Co-zyBench requires installing Python3[3] (version>=3.8) and EnergyPlus (version =22.2.0). Users can execute the following Anaconda[4] command line to create a *cozybench* environment through the *cozybench.yml* configuration file:

```
> conda env create -f path-to-cozybench/env/
    cozybench.yml -n cozybench
```

B. Integrating a Thermal Comfort Provision System

To evaluate a thermal comfort provision system using Co-zyBench, the user needs to integrate their code to the co-simulation middleware. To facilitate this process, we include

[2]https://fmi-standard.org/
[3]https://www.python.org/
[4]https://www.anaconda.com/

sample systems in the `strategies.py` file of Co-zyBench's GitHub repository [3]. Co-zyBench provides the user with parameters extracted from the DTs during the execution of the benchmark that can be used within the thermal comfort provision system (see Table I). In addition to using the parameters as desired, the thermal comfort provision system has to return the computed group thermal comfort value (between -3 to 3) for each room. Then, Co-zyBench implements such value into the HVAC system through the `generate_set_point()` function in `strategies.py` (which can be adapted if desired). For more details, refer to the Cozy-Bench article [1].

Provided Parameters	
thermal_sensation	thermal sensation per occupant
p_loss	accumulated extra loss per occupant
indoor_temp	indoor temperature per room in °C
temp_out	outdoor temperature in °C
ec_cooling_coil	energy consumed by cooling coil in J
ec_heating_coil	energy consumed by heating coil in J
ec_fan	energy consumed by the AHU fan in J

TABLE I: Parameters provided for system design.

C. Setting up a Building DT

This phase involves defining the Building DT. Users can use our reference buildings in the `./model` folder of our repository or customize the building model depending on their requirements. For the customization, EnergyPlusToFMU[5] is required. The initial step in creating building models for evaluation involves configuring the energy model of the building and its HVAC system. After creating the EnergyPlus model, the next step is to define the input and output parameters for generating its FMI model. Examples of the EnergyPlus models of an office building are provided in our GitHub repository [3] along with the following scripts to automate the FMI configuration.

```
> python ep_configure.py path/to/Energy+.idd path/to
    /energyplus_model
```

Now the model is ready for generating the FMI model using `EnergyPlusToFMU` using the following command:

```
> python path/to/EnergyPlusToFMU.py -i /to/Energy+.
    idd  -w /to/weather.epw -a 1 /to/model.idf
```

D. Setting up an Occupant DT

To customize the occupant DT, users must define occupants and events corresponding to their designed building scenarios. This involves identifying potential events and the corresponding attendees. We utilize SmartSPEC for generating such trajectories – for further details refer to SmartSPEC documentation [2].

The preparation also involves creating occupant profiles for individual thermal sensation models. This is achieved by leveraging the ASHRAE Global Thermal Comfort Database II[6] for historical data. A KNN classifier is deployed to predict thermal sensation taking into account occupant gender, age, height, weight and indoor temperature. Users can customize this process with their historical dataset in `./knn` and the preferred parameters.

[5]https://simulationresearch.lbl.gov/projects/energyplustofmu

[6]https://github.com/CenterForTheBuiltEnvironment/ashrae-db-II

E. Running Co-zyBench

To evaluate the performance of a defined thermal comfort provision system, the following command must be executed:

```
> python ./main.py -s system -b building -o occupant
    -p profile
```

Here, *system* specifies the chosen thermal comfort provision system(s), defined as a list in case there are more systems to be evaluated. The optional parameter *-b building* is the path to the building FMI model, *-o occupant* and *-p profile* are for occupant trajectory and defined profiles. The reference scenarios of Co-zyBench are stored in `./models` of our repository. For example, to conduct the evaluations in a predefined office building and occupants scenario included in the benchmark using the Parisian climate zone, users can run the following command:

```
> python ./main.py -s system -b ./models/office/
    Paris/in.fmu -o ./models/office/trajectories -p
    ./models/office/occ_config.txt
```

Co-zyBench outputs the performance results per simulation day in JSON format:

```
{...
"2023-12-29": {
    "cooling_consumption": 975381990.424281,
    "heating_consumption": 6603621637.116568,
    "itc": {
        "1": 736.0,
        "2": 768.0,
        ...},
    "total_itc": 6333.0,
    "equality": {
        "1": -22.526659451659395,
        "2": 62.958621933621814,
        ...}
}}
```

where *cooling_consumption* and *heating_consumption* represent the energy usage of cooling and heating systems in joule. *itc* is the discomfort experienced by each occupant (*total_itc* refers to the total discomfort). *equality* shows how "fair" the system is to each occupant when selecting a temperature to implement. The wider the gap between the values, the more inequitable the system is. In summary, when assessing the results keep in consideration that a system performs best if the energy consumption values and overall discomfort values are small and the equality values are close to zero for all occupants.

REFERENCES

[1] M. Jun, P. Dimitrije, Y. Roberto, and B. Georgios, "Co-zybench: Using co-simulation and digital twins to benchmark thermal comfort provision in smart buildings," in *IEEE International Conference on Pervasive Computing and Communications (PerCom)*, 2024.

[2] https://github.com/andrewgchio/SmartSPEC, SmartSPEC".

[3] https://github.com/satrai-lab/cozybench, Co-zyBench".