Optimisation Of Controller Parameters For Adaptive Building Envelopes Through A Co-Simulation Interface: A Case Study

Esther Borkowski[1], Mattia Donato[2], Giovanni Zemella[2], Dimitrios Rovas[1], Rokia Raslan[1]
[1]Institute for Environmental Design and Engineering, University College London, Central House,
14 Upper Woburn Place, London WC1H 0NN, United Kingdom
[2]Ove Arup & Partners, 13 Fitzroy Street, London W1T 4BQ, United Kingdom

Abstract

Adaptive building envelopes can dynamically adapt to environmental changes, often supported by a control system. While building performance simulation (BPS) tools can be employed to test different design alternatives, representing control strategies within current BPS tools can be challenging, especially for systems with a fast, dynamic response. Another challenge in current BPS tools is the ability to tune and select parameters for the particular use case. In this study, a modelling approach is presented for the integrated analysis of control strategies of adaptive building envelopes linking thermal performance and control with an optimisation algorithm. The proposed modelling approach was evaluated using a case study with an automated motorised blind with two distinct control strategies. Simulation results suggest that the window heat gains were 72.7 % lower when the controller model was coupled with an optimiser to identify optimised controller parameters compared to a baseline control strategy. The results of this study are suggestive of the benefits that can be obtained from adjusting the dynamic aspects of the building envelope. The results support the thesis of using optimisation as standard building envelope design practice in the future.

Introduction

Adaptive building envelopes can dynamically adapt to changes in the external environment with the aim of improving the environmental performance of buildings (Loonen et al., 2016). Their dynamic behaviour is achieved either by dynamics of the building envelope itself, as in case of phase change material (PCM), or by mechanical actuation supported by a control system that generates actuation commands to drive the transition between operational states. Taking into account past, current and possibly forecast conditions, such a controller can determine actions and implement them to adjust building envelope characteristics to improve building performance as measured by relevant performance indicators.

To optimise the use of such controllers, building performance simulation (BPS) tools can be employed to test different design alternatives. A vital aspect of this is the ability to simultaneously simulate the building envelope along with the control strategy to enable an integrated analysis of such interacting building systems (Mazzarella and Pasini, 2009). However, the types and ranges of control strategies that can be modelled in current BPS tools, such as EnergyPlus (National Renewable Energy Laboratory (NREL), 2018), are limited (Widl et al., 2014).

An example of these limitations is the modelling assumptions in current BPS tools that hinder proper modelling and prediction of the influence of control decisions on the dynamic performance of building envelopes. This is evident in the case of time step and state event handling of current BPS tools (Nouidui and Wetter, 2014). EnergyPlus, for example, has a minimum time step of one minute and cannot handle state events, which occur in a simulation if a value in the state variables of a model changes. This hinders EnergyPlus’ ability to simulate models that change dynamically. Another example of these limitations is, according to Favoino et al. (2016), that current BPS tools cannot be used to model control strategies that aim to optimise any selected objectives. Optimisation generally refers to an iterative process of selecting a near-optimal solution to a problem in the search space (Nguyen, Reiter and Rigo, 2014). If an optimisation tool (i.e., algorithm or strategy) is coupled with a BPS in an automated process, the literature tends to use the term simulation-based optimisation. Because current BPS tools are limited in simulation-based optimisations, Delgarm et al. (2016) coupled EnergyPlus through jEPlus (Zhang, 2009; Zhang and Korolija, 2010), an EnergyPlus simulation manager for parametric analyses, with MATLAB (MathWorks, 2018), a high-level programming language, to optimise various design parameters, such as overhang characteristics, to improve the energy performance of four case study buildings. Comparing each case study building to a baseline model, Delgarm et al. (2016) found that the total annual energy consumption was reduced by 23 % to 42 % through the optimisation. Even though these savings by control alone are quite ambitious as indicated in BS EN 15232-1:2017 (RHE/16, 2017) and highly depend on the choice of the baseline model, a well-tuned control strategy may have positive impacts on energy use, but also on thermal performance and occupant comfort (Treado, 2013).

To better address systems or controls with a fast, dynamic response and to better tune and select controller parameters, a modelling approach for the analysis of control strategies of adaptive building envelopes is proposed. This will aim to establish an efficient methodological approach to link the whole-building simulation of thermal performance and control with an optimisation algorithm. Although similar approaches have been considered in the context of HVAC systems, less attention has been paid to adaptive building envelopes. The modelling approach was evaluated through a case study of a closed cavity façade with an automated motorised blind for an office development in London with two distinct control strategies.

Background: simulation-based optimisation

A growing body of literature has recently investigated simulation-based optimisations used to identify optimised controller parameters for adaptive building envelopes with the overall aim of satisfying several conflicting design criteria, such as minimising cost while maximising performance (e.g., Evins, 2013; Attia et al., 2013). To develop an understanding of the application of simulation-based optimisations, Attia et al. (2013) conducted interviews with 28 optimisation experts and found that EnergyPlus and IDA ICE (EQUA Simulation AB, 2018) are the most used BPS tools and that MATLAB and GenOpt (Wetter, 2016a), an optimisation tool for the minimisation of a cost function evaluated by an external BPS tool, are the most widely-used tools for simulation-based optimisations among interviewees.

One of the main challenges for the integration of optimisations into the design process identified by this study is a “lack of awareness and confidence in the use of optimisation” (Attia et al., 2013, p. 3703). The key problem with this finding is that optimisation algorithms offer an effective way to create workflows less prone to human errors, thus increasing simulation capabilities and ultimately guiding the design thanks to established, quick and reliable workflows. As such, research into solving the problem of integrating optimisations into the design process is already underway, and numerous studies have attempted to develop frameworks for the optimisation of controller parameters for adaptive building envelopes. As an example, Favoino, Jin and Overend (2017) proposed a framework for optimising design and control aspects of adaptive insulation. To take account of the simultaneous variation of both design and control aspects, the authors coupled EnergyPlus to predict building performance and MATLAB to perform optimisation and coordinate the process. MATLAB is extensively used for simulation-based optimisations due to the high-level nature of the language that avoids intricacies like active memory management as well as easy interfacing with external tools and robust implementation of optimisation algorithms. Despite its capabilities, a potential shortcoming of MATLAB and other similar programming languages is that they were not explicitly designed for simulation-based optimisations and, as a result, require some programming skills to use that may not be available in design practices (Nguyen, Reiter and Rigo, 2014). A more focused tool developed for simulation-based building optimisations is GenOpt. It offers, however, limited support to address multi-objective optimisation problems. Consequently, researchers examined other tools to identify optimised controller parameters for adaptive building envelopes. This includes GENE_ARCH (Caldas, 2006), an evolution-based generative design environment, which was used, for instance, by Wang and Beltran (2016) to optimise the control sequence for varying window and wall properties, or the Building Controls Virtual Test Bed (BCVTB) (Wetter, 2016b), a software environment traditionally used for co-simulation, which was employed, for example, by Evins, Pointer and Vaidyanathan (2011) to optimise ventilation control of a double-skin façade.

An alternative tool that can be utilised for the optimisation of controller parameters for adaptive building envelopes is Python (Python Software Foundation, 2018), a high-level programming language. Although Python and MATLAB are at first sight very similar, Python has various advantages over MATLAB (Ozgur et al., 2017):

- **Cost.** Python is a free programming language.
- **Portability.** Python can run on all operating systems.
- **Libraries.** Python provides numerous libraries, e.g., for co-simulation and optimisation problems.

Due to the benefits presented above, Python was chosen for this study to perform a simulation-based optimisation to identify optimised controller parameters for an adaptive building envelope. In the literature, only a few investigations that optimise building envelope characteristics or properties through Python were identified. An example of this is a study conducted by Cascone, Capozzoli and Perino (2018), who coupled a building model with PCM implemented in EnergyPlus with an optimisation algorithm written in Python to optimise energy and cost by varying, among others, the thickness and thermo-physical properties of PCM.

Since the work presented in this paper focuses on the optimisation of controller parameters for an adaptive building envelope, it necessitates the use of a tool that offers modular and flexible modelling and simulation methods for control systems, as in case of the Modelica environment Dymola (Dassault Systèmes, 2018; Modelica Association, 2017). Hence, this study extends the optimisation approach by Cascone, Capozzoli and Perino (2018) by coupling EnergyPlus to simulate the thermal dynamics and Dymola to host the control logic through the Functional Mock-up Interface (FMI) standard (MODELISAR, 2010), an open interface for communication and information exchange in co-simulation setups. The FMI standard had been favoured over, e.g., the BCVTB because it is a non-proprietary industry standard that enables model reuse (Wetter, Grahovac and Hu, 2018). Linking this co-simulation with the optimisation algorithm implemented in Python, the modelling approach developed in this study offers the potential to identify optimised controller parameters for adaptive building envelopes.

Methods

To investigate the application of the modelling approach, the research design used in this paper adopted a case study approach due to the exploratory nature of the research (Yin, 2014). A case study approach was also used to yield insights into the operation of control strategies for adaptive building envelopes in their real-life setting mimicked by the simulation. To represent a broader set of cases, a common case was selected, which was an 18 m^2 bay of a typical floor of a 50-storey office development in Central London. The South-oriented closed cavity façade of the case study had an automated motorised blind with two distinct control strategies as the subunits (i.e., subject) of the analysis, as illustrated in Figure 1.

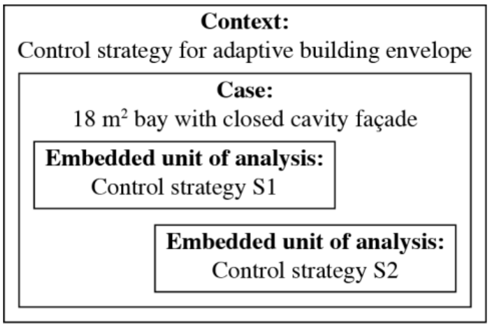
Context:
Control strategy for adaptive building envelope
Case:
18 m² bay with closed cavity façade
Embedded unit of analysis:
Control strategy S1
Embedded unit of analysis:
Control strategy S2

Figure 1: Embedded case study design

A thermal simulation model developed in EnergyPlus was combined in a co-simulation setup with a control model developed in Dymola. To exchange information at each time step between EnergyPlus and Dymola, the EnergyPlus model was encapsulated and shared as a Functional Mock-up Unit (FMU) and imported into Dymola, where the FMU appeared as input/output block to be connected to other models. The data exchange between EnergyPlus and Dymola was ensured through the FMI standard. This integrated model was then wrapped with an optimiser to understand and fine-tune controller parameters and to indirectly select control strategies. The methodology adopted can be subdivided into three main phases: (i) pre-processing, (ii) simulation and optimisation and (iii) post-processing. To automate these phases, this study used Python v3.6.6 with Spyder v3.1.1 (Spyder Project Contributors, 2018), a scientific Python development environment. Python particularly facilitated the integration of Dymola and EnergyPlus with the optimisation package DEAP for controller parameter optimisation. Other commonly used Python packages were BuildingsPy (Lawrence Berkeley National Laboratory (LBNL), 2018) to automatically instantiate and initialise Dymoly simulations, DyMat (Rädler, 2015) to read and process Dymola results files and Matplotlib (Hunter, 2007) to plot and visualise data.

Pre-processing

The pre-processing included the development and structuring of the co-simulation of thermal performance and control and of the optimisation algorithm. The first step was to implement the room model, whose simulation parameters are presented in Table 1, in EnergyPlus v8.9.0. To couple EnergyPlus to another simulation tool, the External Interface of EnergyPlus, whose objects received their inputs from Dymola at each time step, had to be activated. The software package EnergyPlusToFMU v2.0.3 (Nouidui, Lorenzetti and Wetter, 2018) was used to export the case study modelled in EnergyPlus as an FMU for co-simulation using the FMI standard v1.0. This FMU slave was then imported in and executed by Dymola v2019 FD01 as master simulation tool, which supported the import of the FMU for co-simulation and was responsible for coordinating the overall simulation and data transfer.

Table 1: Simulation parameters of room model

Parameter	Condition
Closed cavity façade	Centre-pane U-value: 0.9 W/m^2K g-value: 0.12 - 0.52 Visible light transmittance: 5 - 65 %
Blind	Slat width: 80 mm Slat distance: 72 mm Colour: light grey Solar reflectance: 65 % Visible reflectance: 71 %
Occupancy	07:00 - 19:00
Air change rate	-0.3 h^{-1}
Thermostat setpoints	24 °C

Table 2: Control thresholds for each blind position

Blind position	Control thresholds *
Open	*SolRad* < 240
Horizontal	240 ≤ *SolRad* < 330 AND *SolAlt* > 42.0
Tilt 30°	330 ≤ *SolRad* < 530 AND 42.0 ≥ *SolAlt* > 24.8
Tilt 45°	530 ≤ *SolRad* < 770 AND 24.8 ≥ *SolAlt* > 15.3
Tilt 60°	770 ≤ *SolRad* < 920 AND 15.3 ≥ *SolAlt* > 3.90
Closed	920 ≤ *SolRad AND* 3.9 ≥ *SolAlt*
* *SolRad* – incident solar radiation [W/m^2] *SolAlt* – site solar altitude [°]	

The next step was to set up the model for the blind controller in Dymola to actuate the blind position *yBlind* and the slat angle *ySlat* of the EnergyPlus model. The two control strategies studied were:

- **S1 – Rule-based control strategy as a baseline.** Both *yBlind* and *ySlat* were based on (i) the intensity of the incident solar radiation (*SolRad*) to control solar heat gains and (ii) the site solar altitude (*SolAlt*) to prevent direct sunlight passing through the façade. Control thresholds are apparent from Table 2 and were selected based on the resulting g-value calculated to limit solar heat gains to 100 W/m^2 of floor area based on a 4.5 m floor depth. When the sky was cloudy with a global horizontal illuminance (*HorIll*) lower than 15,000 lux, the control strategy was overridden, and the blind was fully retracted.
- **S2 – Control strategy with optimised controller parameters.** To being able to reuse the baseline model and to simply overwrite *yBlind* and *ySlat* with a set of controller parameters obtained by the

optimisation algorithm, conditional clauses for the components *yBlind* and *ySlat* were added in Dymola to make them mutually exclusive of the components of the baseline model, as shown in Figure 2. Including only necessary components in the simulation helped to reduce the computational weight of calculating results that were not needed or used.

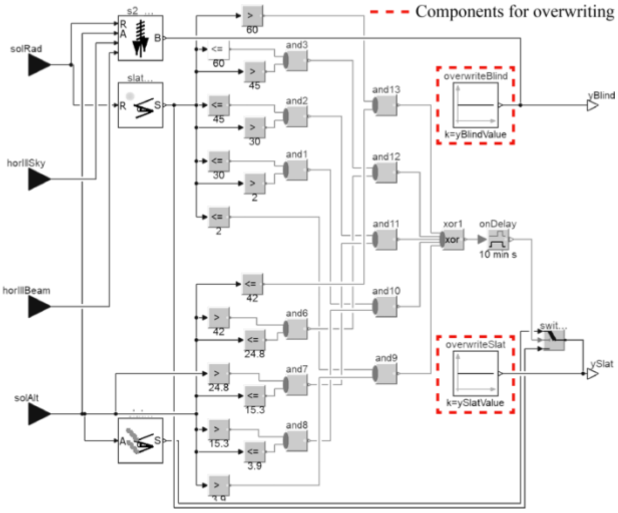

Figure 2: Controller model in Dymola with components to overwrite yBlind and ySlat

After setting up the co-simulation, the optimisation algorithm was written in Python. The optimisation aimed to find the set of values that the controller parameters should adopt to optimise the objective functions. The controller parameters under investigation were the blind position (discrete) and the slat angle (continuous) as can be seen from Table 3.

Table 3: Design variables

Variable	Value
Blind position *yBlind*	0 (open) or 9 (closed)
Slat angle *ySlat*	0° to 90°

To take account of various conflicting criteria that may occur during building operation, such as to improve the quality of human life while maintaining the capacity of the ecosystem, this study used a multi-objective optimisation approach with energy use and thermal comfort as objectives. Given the two-dimensional design variable vector $x = \{X_1, X_2\}$ in the solution space X, the proposed multi-objective optimisation algorithm should find the vector x^* that minimises the given set of objective functions:

$$Z(x^*) = \{Z_1(x^*), Z_2(x^*)\} \qquad (1)$$

The first objective was the minimisation of the total energy use per unit of floor area E_{total} [Wh/m^2] calculated as shown in Equation 2, where $E_{heating}$ and $E_{cooling}$ were the energy uses for space heating and cooling over the period of one simulation time step of 3600 seconds and where A was the conditioned room area.

$$min\ Z_1(x) = E_{total} = (E_{heating} + E_{cooling}) / A \qquad (2)$$

The second objective was the minimisation of the percentage of occupied hours characterised by indoor thermal discomfort DH [%] as shown in Equation 3, where dh was the number of hours characterised by thermal discomfort with an average predicted mean vote (PMV) ± 0.5 and where h was the number of occupied hours.

$$min\ Z_2(x) = DH = dh / h \cdot 100 \qquad (3)$$

The final step of the pre-processing was to execute the setup of the simulation, including simulation parameters (e.g., start/end time and time step) and solver settings (e.g., type and tolerance) with BuildingsPy.

Simulation and optimisation

To optimise the objective functions, the first step of the simulation and optimisation phase was to pseudo-randomly generate an initial population in DEAP based on the aforementioned design variables. As the optimisation algorithm was implemented in Python while the evaluation of the objectives needed the use of Dymola, a data exchange mechanism between Python and Dymola was required. This was achieved through a coupling function written in Python and schematically shown in Figure 3. This coupling function made it possible to:

- convert the multidimensional sets of individuals into a Dymola syntax with the help of BuildingyPy’s *addParameters* function that added parameter declarations to the simulator and allowed to change data on the same simulation model;
- execute the start of the simulations with the updated sets of individuals in batch mode;
- save the Dymola results files to the corresponding result directory; and
- process the Dymola results files in Python to evaluate the objective functions with reference to each individual.

To find a near-optimal solution to the multi-objective optimisation problem, a Pareto-based approach was used. This approach allows examining a set of trade-off optimal solutions from the best-known Pareto set. The set of all nondominated solutions, which represents the set of solutions that is not better than another set of solutions, is called trade-off or Pareto front. A frequently used method to produce a representative subset of the Pareto set is stochastic population-based algorithms, such as the nondominated sorting genetic algorithm (NSGA-II) introduced by Deb et al. (2002). A key advantage of NSGA-II compared to other genetic algorithms, a search heuristic inspired by natural evolution, is that its selection operators had been specially designed to work in multi-objective optimisation problems. For instance, NSGA-II does not select its best individuals through fitness values, but through a combination of values obtained by (i) nondominated sorting, which sorts a population according to the levels of nondominance and identifies nondominated fronts, and (ii) crowding distance algorithms, which guide the selection process to obtain a uniformly spread-out Pareto front.

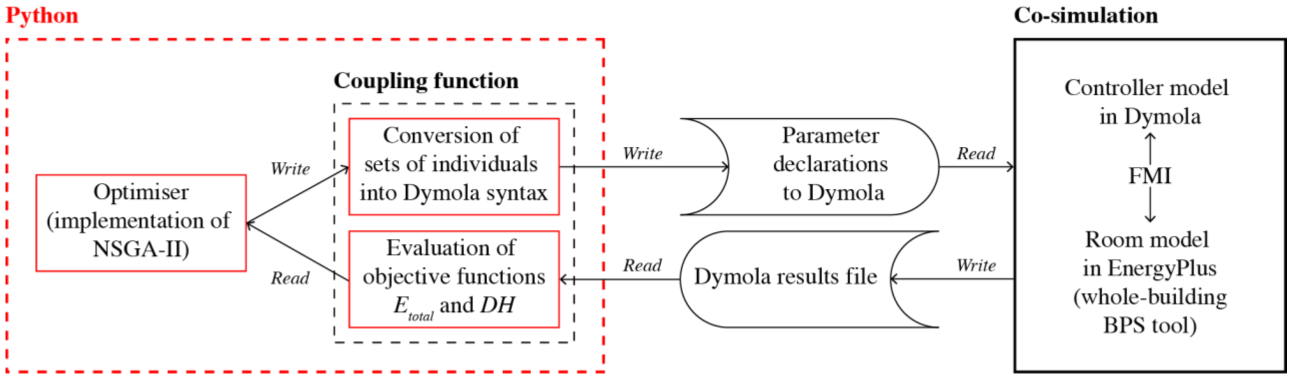
Python
Coupling function
Optimiser (implementation of NSGA-II)
Write
Read
Conversion of sets of individuals into Dymola syntax
Evaluation of objective functions E_{total} and DH
Write
Read
Parameter declarations to Dymola
Dymola results file
Read
Write
Co-simulation
Controller model in Dymola
FMI
Room model in EnergyPlus (whole-building BPS tool)

Figure 3: Coupling between Python and co-simulation with Dymola as co-simulation master

The implementation of the NSGA-II algorithm in the modelling approach through DEAP used the parameters listed in Table 4. These parameters were selected based on Grefenstette's (1986) attempt to determine an ideal set of parameters for genetic algorithms. He found that good performance of genetic algorithms with small populations (20 to 40) can be achieved through "either a high crossover rate combined with a low mutation rate or a low crossover rate combined with a high mutation rate" (Grefenstette, 1986, p. 127). While following Grefenstette (1986) seems to be reasonable for this exploratory study, future research should carry out a convergence test based on the hypervolume indicator that evaluates the performance of multi-objective genetic algorithms by calculating the space covered by solutions in the objective space.

Table 4: NSGA-II parameters

Parameter	Value
Population size	20
Maximum number of generations	5
Crossover probability	70 %
Mutation probability	30 %

For each simulation time step starting on 2 July (15,811,200 seconds), the beginning of a typical three-day summer period, a Pareto front was generated. The fast nondominated sorting approach by Deb et al. (2002) was used to select the "best" control strategy indirectly, and the corresponding Dymola results file was stored for post-processing. The optimisation then moved forward in time to repeat the process for the next time step until the stop time of 16,070,400 seconds was reached. While the controller parameters were optimised per time step, simulations were run for the three-day period as EnergyPlus requires a simulation length of a multiple of a day. Each simulation and optimisation run took approximately thirty minutes on Parallels Desktop 14 for Mac with two cores (2.7 GHz Intel Core i5 processor) and 8 GB memory.

Post-processing

The post-processing aimed to compare the predicted performance of the case study with the two distinct control strategies S1 (rule-based control strategy as a baseline) and S2 (control strategy with optimised controller parameters) by utilising window heat gains and losses, which refer to the total heat flow through a window, as performance indicators. To automatically combine the predicted data of S1 and S2 into one dataset, DyMat was used. The data of the dataset were then statistically analysed and graphically presented and compared by using, among others, Matplotlib.

Results

Simulation results suggest that the control strategies S1 and S2 significantly influenced the predicted performance of the case study. As can be seen in Figure 4, the mean window heat gains of S2 were 72.7 % lower than the mean window heat gains of S1 (S1: 243.3 Wh, S2: 66.4 Wh), especially during the day. Besides, the S2 mean window heat losses were 8.7 % higher than the S1 mean window heat losses (S1: 21.9 Wh, S2: 23.8 Wh).

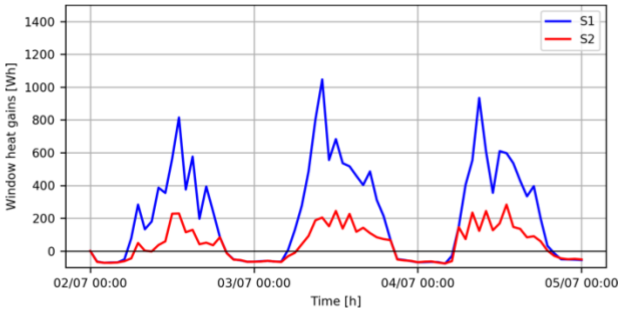

Figure 4: Window heat gains

These differences can be explained by the optimisation's objective function that aimed, among others, at minimising the energy use. The simulation results point out that the energy use could be best minimised with low window heat gains and high window heat losses, which result in a lower cooling load within the building and hence a lower energy use. This was confirmed by the mean energy use for space cooling, which was 24.5 % lower in the case of S2 than of S1 (S1: 522.9 Wh, S2: 394.8 Wh). As illustrated in Figure 5, S2 had lower window heat gain values consistently for each specific value of the energy use than S1.

The results confirm that the modelling approach was able to decrease the energy use for space cooling by optimising controller parameters compared to the baseline control strategy S1. However, objectives not taken into account in the optimisation were energy use for artificial lighting and visual comfort that can be assessed through the key factors of risk of glare and outside viewing possibilities. To allow conclusions to be drawn on energy use for artificial lighting and visual comfort, the percentage of time each blind position occurred during occupied hours was analysed. As shown in Figure 6, the blind was more often open in S1 (47.2 %) than in S2 (0.0 %) indicating that the blind was shut down 52.8 % of the time during occupied hours in S1 and 100 % of the time in S2. Even if the blind was shut down in S1, it occurred less often in tilted or closed position in comparison to S2 (77.8 %).

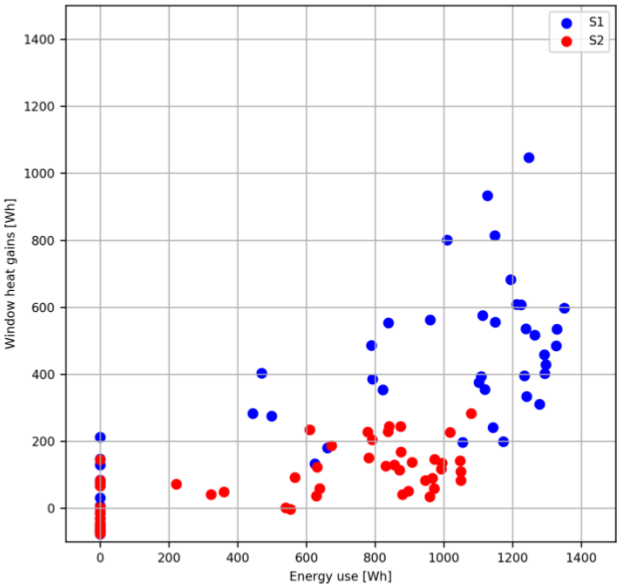

Figure 5: Window heat gains as a function of energy use

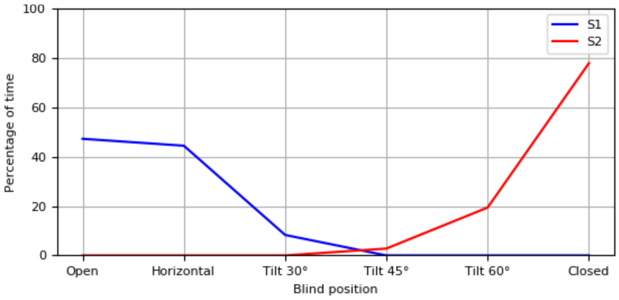

Figure 6: Blind positions during occupied hours

These differences can be again explained by the objective function that could minimise the energy use best when the blind was shut down and more often in tilted or closed position. As no direct sunlight passed through the building envelope in case of S2, it can be assumed, on the one hand, that the energy use for artificial lighting was increased compared to the baseline. On the other hand, the risk of glare might be decreased and visual comfort might thus be improved. On the contrary, the outside view was obstructed since the blind was shut down all the time, which limited occupants' possibilities to look outside and, as a result, might diminish visual comfort. A further study with more focus on energy use for artificial lighting and on well-being of occupants in the presence of daylight, e.g., by constraining the illumination at desk level, will, therefore, be undertaken.

Discussion

The modelling approach developed in this study aimed to identify optimised controller parameters for adaptive building envelopes. Although energy use for artificial lighting and visual comfort were not taken into account, the results of this study support the idea that the developed modelling approach offers benefits that would not be expected from current BPS tools:

- **Capability.** The proposed modelling approach automatically searched for the optimised solutions from a set of available controller parameters, which may have helped to identify automatically and verifiably proper tuning of control to achieve the design objectives.
- **Flexibility and extensibility.** The use of the Modelica environment Dymola led to a high working efficiency, which can be illustrated by the reuse of the controller model for both control strategies studied.
- **Workflow automation.** The modelling approach was entirely automated through a Python script to perform many common repetitive or technically intensive tasks. This may have simplified and speeded up the process of, e.g., simulation input file generation and reduced error rates.

However, the modelling approach has also several possible limitations:

- **Usability.** As mentioned earlier, a potential shortcoming of programming languages, such as MATLAB or Python, is that they were not specifically designed for simulation-based optimisations and, consequently, require high expertise to use. This may lead to diminished confidence in the use of this approach.
- **Efficiency.** A general limitation of simulation-based optimisations is the lack of a standard systematic approach, which may lower the efficiency regarding time and performance improvement. Although this study adopted many existing Python packages to assist in achieving a higher level of efficiency, basic skills in programming, data processing and statistical analysis are required.
- **Coupling barrier.** Missing interfaces that integrate and link BPS tools and optimisation seamlessly may inhibit the use of simulation-based optimisations in practice. While this study used, for instance, the FMI standard as a non-proprietary industry standard, it may be possible that not all BPS tools support the FMI standard, which may impose a coupling barrier on the modelling approach.

Notwithstanding these limitations, the present results add to the growing body of literature on simulation-based optimisations in at least two major respects. Firstly, the study investigated the development of a simulation method that coupled an optimisation approach with a BPS tool to simulate the thermal dynamics. Following this, a novel environment to host the control logic was established to determine the energy saving potential of adaptive building envelopes. An implication of this is that the proposed modelling approach has the potential to overcome the difficulties of trial-and-error typically associated with stand-alone simulations. Secondly, this work highlighted a process that used co-simulation to generate a detailed parametric model of adaptive building envelopes, with key parameters being controller parameters. This parametric model can then be invoked by optimisation tools to explore parameter search spaces and to identify automatically and verifiably proper tuning of control to achieve design objectives. Therefore, the proposed modelling approach could potentially contribute to the use of optimisation methods as standard practice within the design of adaptive building envelopes in the near future.

Conclusion

This study aimed to evaluate the proposed methodological approach through a case study that included an automated motorised blind with two distinct control strategies, where optimisation was utilised to identify optimised controller parameters using a stochastic population-based algorithm to fine-tune and select controller parameters. In the light of the exploratory nature of this study, the controller parameters were optimised over just a typical three-day summer period. The results of this evaluation show that the mean window heat gains and the mean energy use for space cooling of the control strategy with optimised controller parameters were considerably lower than those of the baseline control strategy. Since these results are suggestive of the benefits that can be obtained from adjusting the dynamic aspects of the building envelope and that are only partly supported by current BPS tools, they have two important implications for future research:

- The present objective function only aims to minimise the energy use and the number of hours characterised by thermal discomfort. This results in a control algorithm that keeps the blind shut down for most of the day even though the office is occupied possibly affecting the well-being of occupants as well as the energy use for artificial lighting. Consequently, the first implication of the results is to undertake further studies to take account of energy use for artificial lighting and visual comfort to keep the blind more often open during occupied hours.
- While the results of this study provide some quantitative evidence that the proposed modelling approach can find the “optimal” controller parameters for the three-day period, it might be possible that the studied control strategy is not the best when looking at longer periods, i.e., the whole cooling season. It might, for instance, be better to keep control parameters constant or to follow the model-predictive control (MPC) paradigm and change the parameters on a more regular basis. As a result, a second implication of the study results is that various control strategies will be investigated over longer periods in the future to establish validity and reliability of the control algorithm.

References

Attia, S., M. Hamdy, W. O’Brien and S. Carlucci. (2013). Assessing gaps and needs for integrating building performance optimization tools in net zero energy buildings design. *Energy and Buildings 60*, 110–124.

Caldas, L. (2006). GENE_ARCH: An Evolution-Based Generative Design System for Sustainable Architecture. *Intelligent Computing in Engineering and Architecture, 13th EG-ICE Workshop 2006*. Ascona (Switzerland), 25-30 June 2006.

Cascone, Y., A. Capozzoli and M. Perino. (2018). Optimisation analysis of PCM-enhanced opaque building envelope components for the energy retrofitting of office buildings in Mediterranean climates. *Applied Energy 211*, 929–953.

Dassault Systèmes. (2018). Dymola v2019 FD01. Lund (Sweden). Accessed 27 November 2018. www.dymola.com.

De Rainville, F.-M., F.-A. Fortin, M.-A. Gardner, M. Parizeau and C. Gagné. (2012). DEAP: A Python Framework for Evolutionary Algorithms. *GECCO’12*. Philadelphia, PA (USA), 7-11 July 2012.

Deb, K., A. Pratap, S. Agarwal and T. Meyarivan. (2002). A Fast and Elitist Multiobjective Genetic Algoritm: NSGA-II. *IEEE Transactions on Evolutionary Computation 6 (2)*, 182–197.

Delgarm, N., B. Sajadi, S. Delgarm and F. Kowsary. (2016). A novel approach for the simulation-based optimization of the buildings energy consumption using NSGA-II: Case study in Iran. *Energy and Buildings 127*, 552–560.

EQUA Simulation AB. (2018). *IDA ICE v4.8*. Solna (Sweden). Accessed 26 November 2018. https://www.equa.se.

Evins, R. (2013). A review of computational optimisation methods applied to sustainable building design. *Renewable and Sustainable Energy Reviews 22*, 230–245.

Evins, R., P. Pointer and R. Vaidyanathan. (2011). Multi-objective optimisation of the configuration and control of a double-skin facade. *Proceedings of Building Simulation 2011: 12th Conference of International Building Performance Simulation Association*. Sydney (Australia), 14-16 November 2011.

Favoino, F., F. Fiorito, A. Cannavale, G. Ranzi and M. Overend. (2016). Optimal control and performance of photovoltachromic switchable glazing for building integration in temperate climates. *Applied Energy 178*, 943–961.

Favoino, F., Q. Jin and M. Overend. (2017). Design and control optimisation of adaptive insulation systems for office buildings. Part 1: Adaptive technologies and simulation framework. *Energy 127*, 301–309.

Grefenstette, J. (1986). Optimization of Control Parameters for Genetic Algorithms. *IEEE Transactions on Systems, Man, and Cybernetics SMC-16(1)*, 122-128.

Hunter, J.D. (2007). Matplotlib: A 2D graphics environment. *Computing In Science & Engineering 9(3)*, 90–95.

LBNL. (2018). *BuildingsPy v1.7.0*. Berkeley, CA (USA). Accessed 26 November 2018. http://simulationresearch.lbl.gov/modelica/ buildingspy/.

Loonen, R.C.G.M., F. Favoino, J.L.M. Hensen and M. Overend. (2016). Review of current status, requirements and opportunities for building performance simulation of adaptive facades. *Journal of Building Performance Simulation 10(2)*, 1–19.

MathWorks. (2018). *MATLAB R2018b*. Natick, MA (USA). Accessed 21 November 2018. https://www.mathworks.com.

Mazzarella, L., and M. Pasini. (2009). Building Energy Simulation and Object-Oriented Modelling: Review and Reflections Upon Achieved Results and Further Developments. *Eleventh International IBPSA Conference*. Glasgow (Scotland), 27-30 July 2009.

Modelica Association. (2017). *Modelica v3.4.* Linköping (Sweden). Accessed 16 January 2019. https://www.modelica.org.

MODELISAR. (2010). *Functional Mock-up Interface (FMI) v1.0*. Accessed 20 August 2018. http://fmi-standard.org.

Nguyen, A.-T., S. Reiter and P. Rigo. (2014). A review on simulation-based optimization methods applied to building performance analysis. *Applied Energy 113*, 1043–1058.

Nouidui, T.S., D. Lorenzetti and M. Wetter. (2018). *EnergyPlusToFMU v2.0.3*. Berkeley, CA (USA). Accessed 16 January 2019. http://simulationresearch.lbl.gov/fmu/EnergyPlus/export/index.html.

Nouidui, T.S., and M. Wetter. (2014). Tool coupling for the design and operation of building energy and control systems based on the Functional Mock-up Interface standard. *Proceedings of the 10th International Modelica Conference*. Lund (Sweden), 10-12 March 2014.

NREL. (2018). *EnergyPlus v8.9.0*. Lakewood, OH (USA). Accessed 20 August 2018. https://energyplus.net.

Ozgur, C., T. Colliau, G. Rogers, Z. Hughes and E.B. Myer-Tyson. (2017). MatLab vs. Python vs. R. *Journal of Data Science 15*, 355–372.

Python Software Foundation. (2018). *Python v3.6.6*. Wilmington, DE (USA). Accessed 21 November 2018. https://www.python.org.

Rädler, J. (2015). *DyMat v0.7*. Potsdam (Germany). Accessed 3 December 2018. https://bitbucket.org/jraedler/dymat.

RHE/16. (2017). *BS EN 15232-1:2017 - Energy Performance of Buildings. Impact of Building Automation, Controls and Building Management. Modules M10-4,5,6,7,8,9,10.*

Spyder Project Contributors. (2018). *Spyder v3.1.1*. Cambridge, MA (USA). Accessed 21 November 2018. https://www.spyder-ide.org.

Treado, S. (2013). The Role of Optimization in the Design and Operation of Building Control Systems. *ASHRAE Transactions* 119(1), 1-8.

Wang, J.J. and L. Beltran. (2016). Energy Performance Of Future Dynamic Building Envelopes. *Proceedings of the 3rd IBPSA-England Conference BSO 2016*. Newcastle (UK), 12-14 September 2016.

Wetter, M., M. Grahovac and J. Hu. (2018). Control Description Language. *1st American Modelica Conference*. Cambridge, MA (USA), 9-10 October 2018.

Wetter, M. (2016a). *GenOpt v3.1.1*. Berkeley, CA (USA). Accessed 26 November 2018. https://simulationresearch.lbl.gov/GO/.

Wetter, M. (2016b). *BCVTB v1.6.0*. Berkeley, CA (USA). Accessed 27 November 2018. https://simulationresearch.lbl.gov/bcvtb/FrontPage.

Widl, E., B. Delinchant, S. Kübler, D. Li, W. Müller, V. Norrefeldt, T.S. Nouidui et al. (2014). Novel simulation concepts for buildings and community energy systems based on the Functional Mock-up Interface specification. *2014 Workshop on Modeling and Simulation of Cyber-Physical Energy Systems (MSCPES)*. Berlin (Germany), 14 April 2014.

Yin, R. (2014). *Case Study Research: Design and Methods*. SAGE Publications. Thousand Oaks, CA (USA).

Zhang, Y. (2009). Parallel' EnergyPlus and the development of a parametric analysis tool. *Eleventh International IBPSA Conference*. Glasgow (Scotland), 27-30 July 2009.

Zhang, Y. and I. Korolija. (2010). Performing complex parametric simulations with jEPlus. *SET2010 - 9th International Conference on Sustainable Energy Technologies*. Shanghai (China), 24-27 August 2010.