

Stochastic Simulation and Inference using Modelica

Gregory Provan Alberto Venturini
Department of Computer Science,
University College Cork, Cork, Ireland
g.provan, a.venturini@cs.ucc.ie

Abstract

The physical modelling and simulation of systems with inherent uncertainty still poses significant issues when using Modelica and its tools. At present, both language and tools are fundamentally deterministic and offer limited support for handling uncertainty; this limits the scope of using Modelica in certain domains, e.g. feedback control systems. We propose a framework for incorporating uncertainty in Modelica simulation and analysis tasks. We do this by coupling a Modelica model with exogenous stochastic models. Finally, we apply this approach to the domain of building modelling.

Keywords: simulation; stochastic modeling; energy systems modeling

1 Introduction

Physical-model simulation using Modelica has traditionally been viewed as a deterministic problem, despite major sources of uncertainty. This uncertainty arises due to issues such as:

initial conditions incomplete input observations, measurement error, shortcomings in the data assimilation cycle, etc.

model accuracy and fidelity incomplete knowledge of physical processes (e.g., inaccurate parameterizations of sub grid-scale processes). incomplete and inaccurate numerical schemes,

At present, Modelica tools (e.g., Dymola) enable variability of initial conditions by different instantiations of model parameters Θ or by assigning values to internal model variables. However, this assignment can be done only once for each simulation. For simulations in which stochastic variables exist or there are external processes providing data (e.g., sensor/actuator data) to the model on a regular basis, the simulation must be re-started for each new input. This limits the

scope of using Modelica for use with certain feedback control systems (e.g., Model-Predictive control) or in embedded systems.

Consider the case in which Modelica currently deals with stochastic inputs, e.g., if we were to specify a probability distribution (pdf) over Θ . In this case, Monte Carlo (MC) sampling can be used to define a set of initial conditions for simulation. The drawback to this approach is that, for a complex pdf, a large number of samples (and hence simulations) will be needed in order to achieve a stochastically-sound set of simulations.

Throughout this article we will use the domain of energy modeling to explain our concepts. In particular, we will focus on the modeling of buildings, for which there exist several Modelica libraries, e.g., [13], for generating models for large, complex systems.

Our objective is to define a stochastic state evolution approach that is computationally efficient and can make use of existing Modelica deterministic simulators. We propose a framework for incorporating uncertainty in simulation and analysis tasks which use Modelica models. Our contributions are as follows:

- We propose a framework for ensemble-based stochastic optimisation, using Modelica as a deterministic modeling language and simulation methodology.
- We apply this approach to the domain of renewable energy in terms of underfloor heating optimisation.

Our approach shows how one can extend the existing Modelica language and toolset for such tasks. However, it also highlights deficiencies in Modelica for stochastic representation, as well as deficiencies in the Modelica tools to incorporate stochastic inference *within* a simulation, as well as the inability to accept exogenous inputs during a simulation.

2 Related Work

This work aims to extend both Modelica and Building Performance Simulation (BPS) with stochastic methods, and we discuss prior work in both areas.

Little work has focused on stochastic methods in Modelica. Most recently, Bouskela et al. [1] have described (a) methods for stochastic analysis and (b) proposals for identifying stochastic Modelica variables and performing appropriate inference. [11] discusses how a Modelica model can be used as a simulation model within computational design, such that the probability of a feasible design is explicitly computed.

In the area of BPS, Jacob et al. [9] integrate Monte Carlo sampling within embedded optimization for BPS. In particular, they use conditional probability density functions for energy consumption and demand to quantify the difference between a base case (of energy usage) with scenarios in the presence of uncertainty.

[8] shows how uncertainty analysis can improve BPS through a case study of an office building with respect to various building performance parameters, demonstrating the implications of uncertainty in results concerning energy consumption (annual heating and cooling) and thermal comfort (weighted over- and underheating hours).

One in-depth analysis of the impact of uncertainty in BPS, covering notions of *internal* and *external* probabilistic approaches to quantifying the overall effect of parameter uncertainty in building simulations, has been performed [12]. He quantifies the effects of uncertainty in building simulation by considering the internal temperature, annual energy consumption and peak loads. [3] study the potential impact of climate change on current building designs by examining future climates. They employ two methods, mathematical transformations of observed weather (morphing), and synthetic weather generator, to generate future weather files (on an hourly time scale) which are representative of possible future climates. [10] study how exogenous stochastic processes (e.g., meteorological events) influence building thermal processes, and how endogenous (building-internal) process knowledge (e.g., occupancy patterns) can lead to improved building operation.

3 Simulation Framework

We consider an optimisation framework in which our task is to optimise an objective function \mathcal{J} subject to

a set of constraints over the model, $\chi(\Phi)$. For example, we may want to define an optimal controller for controlling the heating system in a building.

We assume that the model Φ_P that we are simulating requires a set of inputs generated by an exogenous stochastic process Φ_O . For example, in building energy simulation, a model Φ_P consists of the building itself, e.g., the building envelope with internal zones, climate control equipment such as HVAC and sensors/actuators, etc.

We partition the variables in a Modelica model Φ_P as $\mathcal{V} = \mathcal{V}^P \cup \mathcal{V}^O$, where \mathcal{V}^P denotes the endogenous variables and \mathcal{V}^O denotes the exogenous variables. Endogenous variables \mathcal{V}^P do not depend (at least, not directly) on any exogenous stochastic process: at each simulation step, they are deterministically calculated by the Modelica solver. By contrast, exogenous variables \mathcal{V}^O depend directly on exogenous inputs, which change over time due to the exogenous stochastic process Φ_O . Therefore, the values of \mathcal{V}^O must be updated every time the stochastic process Φ_O produces new inputs.

For example, the exogenous variables might be weather variables that provide a set of input conditions for weather for a Modelica simulation of Φ_P . In fact, the existing building library [13] has inputs for up to 30 weather variables, such as temperature, wind-speed, etc.

Figure 4 shows an example of a discrete-time simulation process with exogenous model inputs \mathcal{V}_t^O at each time step t . The exogenous model Φ_O performs inference independent of the Modelica simulation, and provides an input for variables \mathcal{V}_t^O at each time step. The Modelica simulation uses these inputs to conduct its simulation. A key insight into this process is that the Modelica model Φ_P must provide inputs of its endogenous variables \mathcal{V}_t^P to the simulation at time $t + 1$, since the model would otherwise take (incorrect) default values for \mathcal{V}_{t+1}^P .

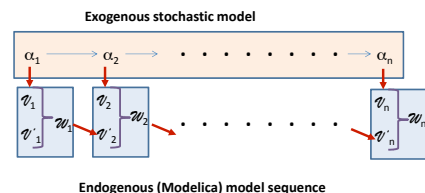


Figure 1: Simple schematic of simulation process with exogenous model inputs

We formalise this process as follows. We define our simulation system as consisting of two models: (a) an

exogenous (possibly stochastic) model Φ_O with variables α , of which a subset \mathcal{V}^O are output variables; (b) a deterministic (endogenous) model Φ_P with variable set \mathcal{V} partitioned into input variables \mathcal{V}^O and internal variables \mathcal{V}_P , and parameters Θ (which are constants over a simulation).

3.1 Stochastic Model Analysis

We further assume an exogenous model \mathcal{E} defined by $\Theta = \psi(\alpha)$ that generates the parameter assignment $\hat{\theta}$. If \mathcal{E} is stochastic, then we have $Pr(\Theta) = Pr(\alpha)$, which defines the joint distribution over α .

We assume a two-step process for model analysis. First, we perform exogenous analysis, which takes the joint set of stochastic inputs α , and through Monte-Carlo (MC) sampling, generates an ensemble of predictions for the parameter set α . Second, we run a simulation for each element of the ensemble, generating an ensemble \mathcal{S} of simulation outputs. Finally, we perform some analysis of the ensemble \mathcal{S} to compute our objective.

3.2 Stochastic Simulation Process

[5] define a *good* probabilistic simulation/forecast as the process of maximizing the accuracy of the predictive distributions subject to calibration, where *accuracy* refers to the spread of the predictive distributions, and is a property of the forecasts only. *Calibration* is the statistical compatibility between the predictive simulation output (or distributions for stochastic models) and the observations. This is a joint property of the forecasts and the observations. We can jointly assess calibration and accuracy by using proper scoring rules, such as the logarithmic score or the continuous ranked probability score [6].

For example, a proper scoring rule is a function $s(\zeta, x)$ that assigns a numerical score to each pair (ζ, x) , where ζ is the predictive distribution and x is the verifying observation.

3.3 Simulation Analysis

Given a set of n possible input streams to Φ_P , we run n simulations. The key is to now use these n simulations to solve our tasks in order to optimise \mathcal{J} .

Consider the case where we aim to compute an optimal control that optimises \mathcal{J} . Given the n simulations, we want to compute a robust control u^* .

Robust control methods are designed to function properly (e.g., maintain stability) under the condition

that uncertain parameters or disturbances are within some (typically compact) set. For example, this may include the assumption of bounded modelling errors. In contrast with adaptive control (which can adapt to changes in environmental conditions or measurements), robust control methods are static.

In our case, we assume that the MC sampling provides a statistically sound set of simulation conditions. Given that, we can either optimise the worst-case outcome, or optimise within the bounds to the input ensemble.

4 Application Domain: Energy Modeling

4.1 Building Simulation

For the analysis and prediction of the dynamic behavior of building performance indicators such as energy consumption and thermal comfort, building performance simulation (BPS) is a key enabling technology. Previous work has shown that the use of BPS is mostly limited to building design and for checking code compliance for the detailed design [8].

BPS makes a number of assumptions that violate realistic building characteristics. For example, almost all BPS model variables are assumed to be deterministic, even though they are uncertain, due to uncertainty in material characteristics and to external and internal condition changes over time. For example, a BPS model contains a range of internal parameters that are only known imprecisely, e.g., wall/ floor/ ceiling heat-transfer parameters. In addition, this type of model behaves differently based on the building occupancy and usage, both of which change over time.

4.2 Incorporating Weather Forecasts

Today, the preferred method of probabilistic weather prediction is based on ensembles of Numerical Weather Prediction (NWP) forecasts. In this case, each ensemble member is a single-valued, deterministic forecast from an NWP model, i.e., a simulation of an NWP model. The forecasts differ from each other with respect to the two major sources of uncertainty: (1) initial conditions and/or (2) model formulation. Figure 2 shows an example of an ensemble of 11 pressure predictions over time.

The ensemble of forecasts must be post-processed in order to provide an interpretable, single forecast. In other words, statistical post-processing aims to gener-

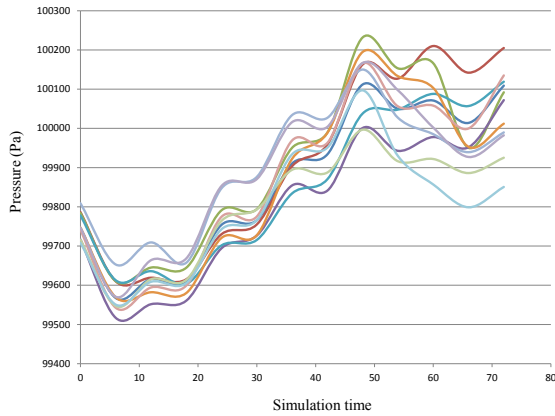


Figure 2: Weather ensemble of 11 pressure predictions over time

ate a calibrated, sharp predictive distribution from the output of NWP ensembles. Two general approaches to the statistical post-processing of forecast ensembles have emerged, namely

- Bayesian model averaging (BMA) [7], where each ensemble member is associated with a kernel function, with a weight that reflects the member’s relative accuracy.
- ensemble model output statistics (EMOS) [4] or nonhomogeneous Gaussian regression (NGR), which fits a single, parametric predictive PDF using summary statistics from the ensemble.

Consider an ensemble forecast, $\lambda_1, \dots, \lambda_m$, for surface temperature, T , at a given time and location. BMA employs Gaussian kernels with a linearly bias-corrected mean: the BMA predictive PDF is the Gaussian mixture with mean \mathcal{N} and variance σ^2 .

$$p(T|\lambda_1, \dots, \lambda_m) = \sum_{i=1}^m w_i \mathcal{N}(a_i + b_i \lambda_i, \sigma^2),$$

with the BMA weights w_1, \dots, w_m , bias parameters a_1, \dots, a_m and b_1, \dots, b_m , and a common spread parameter σ^2 .

The major drawback to this current ensemble approach to physical simulation is that it only applies to single variables, at single locations and single look-ahead times. A key objective in this area is to compute physically consistent probabilistic forecasts of spatio-temporal simulation trajectories.

4.3 Example: Underfloor Heating Example

Consider the case where we can to compute a control setting for the underfloor heating in a zone Z , where

we have uncertainty over the weather forecast and the occupancy for the following day.

We apply our approach to the optimisation of under-floor heating control. Our task is to compute the time interval I during which we “charge” (or heat up) the underfloor slab during the night, such that we jointly maximise user comfort (U_c) and minimise energy usage (U_e) over the following day. Figure 3 depicts a simple example of an underfloor heating system for a house.

We can formulate this task by defining \mathcal{J} as the weighted sum of user comfort and energy usage, with corresponding weights w_c and w_e :

$$\mathcal{J} = w_c U_c + w_e U_e \tag{1}$$

subject to

$$U_c \geq U_c^*, U_E \geq 0 \tag{2}$$

$$\chi(\Phi) \text{ are satisfied} \tag{3}$$

Figure 3: Simple schematic of underfloor heating system for a house

4.4 Underfloor Heating with Stochastic Forecasting

This section describes our underfloor heating model that incorporates stochastic forecasts for weather and occupancy. Figure 4 depicts the variation in tempera-

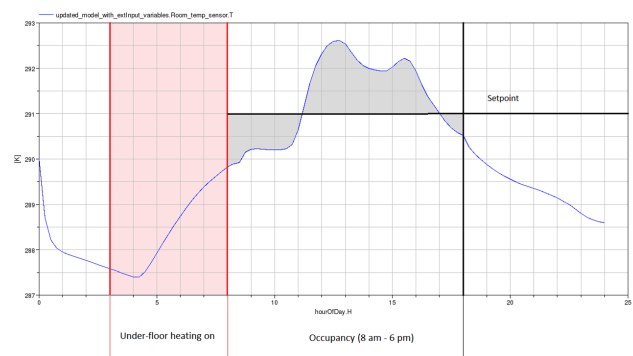


Figure 4: Simulation process for under-floor heating system. The red area shows the hours during which the heating is on. The gray area shows the difference between setpoint and room temperature during office hours.

ture over a day, given that the underfloor heating system is switched on for the period 3–8 am. In this example the temperature set-point for the day is 18° Celsius,

and our objective is to maintain this temperature as closely as possible, in order to optimise the occupants' comfort. The gray area between the actual temperature during the day and the set-point is used to compute a discomfort index, i.e., it is the area denoting the failure to maintain the set-point.

We employ three different models for this application:

- a stochastic model for weather variable prediction;
- a stochastic model for occupancy prediction;
- a Modelica model for simulating the occupied zone in a building with underfloor heating, given as inputs the weather forecasts and the predicted occupancy.

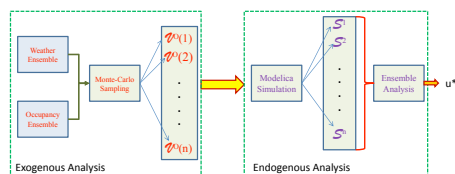


Figure 5: Computational architecture for analysis of underfloor heating system.

Figure 5 depicts our computational architecture, showing the two phases of exogenous computation, where we generate ensembles for weather and occupancy forecasts, and endogenous computation, where we create an ensemble of simulations based on the input ensembles, and then compute the control output u^* that optimises our objective function \mathcal{J} , given the simulation ensembles.

5 Implementation

We have partially implemented the computational architecture described in the previous section. In this section we provide implementation details on our energy simulation model and its inputs, as well as how we intend to use the model for computing an optimal control action u^* .

5.1 Room model

We model a room of one of the buildings on our university campus. This room is an open-space office with a maximum capacity of 12 occupants. The room is equipped with typical office furniture (desks, computers, printers, etc.). The only heating system is

under-floor heating. Additionally, the room has 8 windows and 2 doors. The room is also equipped with sensors that monitor temperature, presence, and luminance.

We model this room by using the Buildings library, developed by Wetter et al. [13]. Figure 6 contains a graphical representation of our model. The main components are:

1. a room component, which extends Buildings.Rooms.MixedAir;
2. an external weather file;
3. heat gains based on occupancy and equipment in the room;
4. an under-floor heating component.

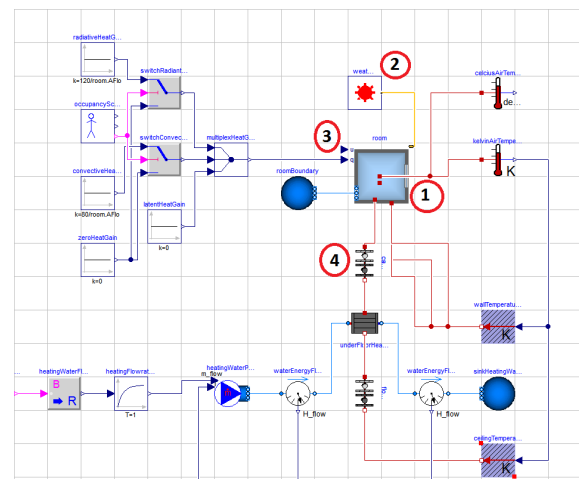


Figure 6: Room model with under-floor heating

5.2 Weather forecast ensembles

Weather is one of the main inputs to our model. Weather data can be either from the past (historical weather records) or in the future (weather forecasts). Since our objective is to implement a control framework, we are interested in weather forecasts. In this section we discuss how we obtain and process weather forecasts.

As mentioned previously, probabilistic weather forecasts are based on ensembles. These ensembles are generated routinely by various data centers around the world. In particular, we use weather forecasts generated by the Global Ensemble Forecast System (GEFS) model [2], which is developed and run by the National Oceanic and Atmospheric Administration (NOAA) in the United States.

As the name suggests, the GEFS is a global model, i.e. it produces forecasts for the whole planet. These forecasts are available for download free-of-charge from the NOAA file servers. The GEFS model produces forecasts up to 16 days in advance; however, since the accuracy tends to degrade quickly, we consider only the first 7 days of prediction. For these first 7 days, the model provides a spatial resolution of 1 degree latitude by 1 degree longitude, and a temporal resolution of 6 hours. The GEFS produces 20 ensemble members. Each member contains the trajectories of various weather variables, e.g. temperature, humidity, pressure, etc.

Our goal is to use these ensemble forecasts to generate probabilistic weather inputs for our model. In order to accomplish this, after downloading the forecast files, we need to carry out a series of steps:

1. forecasts must be spatially interpolated to the point of interest;
2. forecasts must be temporally interpolated;
3. the weather variables that are relevant to our model must be extracted from the forecasts; additionally, some weather variables required by the model are not directly included in the forecasts (e.g. direct and diffuse solar radiation), and therefore must be calculated from the information that is available;
4. the extracted and calculated variables must be statistically post-processed, in order to provide probability distributions;
5. the probability distributions calculated above must be sampled (e.g. by using Monte Carlo methods) to provide weather scenarios;
6. finally, for each sampled scenario, a weather file in the format required by the model must be produced.

We have developed software that performs the above steps, with the exception of the statistical post-processing. At the moment of this writing, instead of generating probability distributions and then sampling from those, we simply create 20 different weather files for each of the 20 ensemble forecasts generated by the GEFS; then, we provide these weather files as inputs to our model. Figure 7 depicts the steps we have implemented to provide weather input to our model.

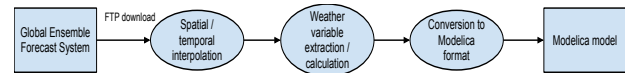


Figure 7: Steps to provide weather input to the room model

5.3 Sensitivity Analysis

The Modelica Buildings library accepts weather files specified as tables of n rows and 30 columns. Each row contains weather conditions for a specific time step, and each column contains the values of a weather variable, specified as real numbers. However, each weather variable has a different impact on the model output (i.e., the room temperature): some variables affect the room temperature more strongly than others; some variables do not affect the room temperature at all. Hence, it is important to precisely assess the impact of each weather variable on the model output, so that only relevant variables need to be extracted from the forecasts.

In this context, we have performed sensitivity analysis on our model in the following way. First, we ran a baseline simulation with a weather file containing historical data. Then, we altered each weather variable in the file by increasing and decreasing its values by 10%, 20% and 30%. Each variable was altered independently of the others; i.e., when we altered one variable, all other variables retained their original values.

For each weather variable, we generated 6 simulations corresponding to the variations in the range of $\{-30\%, -20\%, -10\%, 10\%, 20\%, 30\%\}$. We measured the error between the baseline and each variation; the error was calculated as the integral of the difference of room temperature. The error provides an indication of how much a variable affects the simulation output, with higher error corresponding to higher sensibility.

The results indicate that the model is most sensitive to the following weather variables:

1. dry bulb temperature;
2. direct, diffuse and global solar radiation;
3. opaque sky cover;
4. wind speed and direction;
5. dew point.

Therefore, only these variables are extracted from the GEFS weather forecasts and provided to the model.

5.4 Simulation process

We use the model to address the task of optimal under-floor heating control. Since under-floor heating is a slow-response system, it is normally turned on at night: it is at this time that the concrete slabs are “charged” with heat, which will then be released in the room over the following day. In this context, the output of the control task is u^* , i.e. the amount of hours during which the under-floor heating will be turned on.

In order to calculate u^* we simulate 9 different scenarios S_u , where we vary the amount of under-floor heating hours u from 0 to 8. For each S_u , we calculate $\mathcal{J}_u = w_c U_c + w_e U_e$. Finally, we calculate $u^* = u_{arg \max} \{ \mathcal{J}_u \}$. The value u^* is then given as input to a Building Management System (BMS) which opens the under-floor heating valves for the required amount of time.

This process is repeated every day, for instance at 10 pm. In other words, every 24 hours we run a new series of simulations and we calculate a new control action u^* based on weather (and, potentially, occupancy) forecasts for the day after. However, each time we run a new series of simulations, we cannot reset the model variables to pre-defined initial values. In fact, as stated in section 3, the endogenous variables must be initialized with the values of the previous simulation, whereas the exogenous variables must be initialized according to the external stochastic processes. Using Dymola, this means that the *dsin.txt* file (which provides initial values to variables) must contain the final values of the simulation which generated u^* 24 hours before. This process is depicted in figure 8.

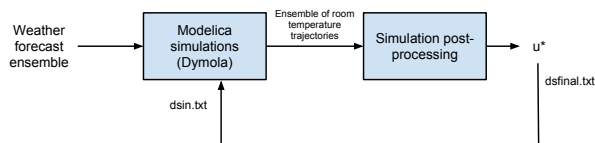


Figure 8: Steps to compute u^* every 24 hours. The post-processing step contains the logic to calculate u^* . The file *dsfinal.txt* contains the final values of the previous simulation which generated u^* .

5.5 Preliminary control results

In order to test our control approach, we first applied it to historical weather data. Our goal here is to compare our control strategy with a 5-hour fixed-schedule strategy, in terms of user comfort and energy usage. We use the simulation process described in the previ-

ous section; the only change is that we use historical weather data instead of forecasts.

We use a set-point of 23 degrees Celsius for week days, and 16 degrees Celsius for weekends. The fixed schedule strategy operates the under-floor heating for 5 hours every night, between 3 am and 8 am. Moreover, it does not differentiate between weekdays and weekends¹. Our control strategy, instead, tries to minimize the error between set-point and room temperature, and thus will tend to turn off the under-floor heating during weekends, when the set-point is lower.

Figure 9 shows the average room temperature obtained with our control strategy (blue trajectory) and the fixed schedule strategy (red trajectory). Although there is some amount of error for both strategies, it is clear that, on average, our control strategy performs better, i.e. it is closer to the desired set-point. Possible ways to further improve our control strategy might consist in (1) leaving the set-point unchanged over weekends (thus avoiding the cooling down of building materials, at the expense of higher energy usage), and (2) extending the possible number of under-floor heating hours to 9 or 10 (at the moment we keep the maximum number of hours to 8).

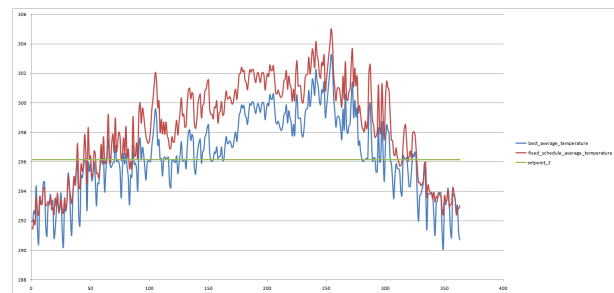


Figure 9: Comparison of average temperature obtained with our control strategy (blue) and a 5-hour fixed schedule strategy (red). The red line represents the desired set-point. The horizontal axis represents days and the vertical axis represents degrees Kelvin.

Figure 10 compares the energy usage of the two strategies. It is apparent that, over the whole year, our control strategy requires significantly less energy than the fixed schedule strategy. This is mainly due to (1) savings during weekends, and (2) savings during the summer season, when the under-floor heating is not needed. It is worth mentioning that, within the model, the energy usage is calculated as the amount of energy (in Joule) that is required to heat up the water which will flow into the under-floor heating pipes.

¹It is worth noting that this control strategy was actually implemented on the building on our university campus.

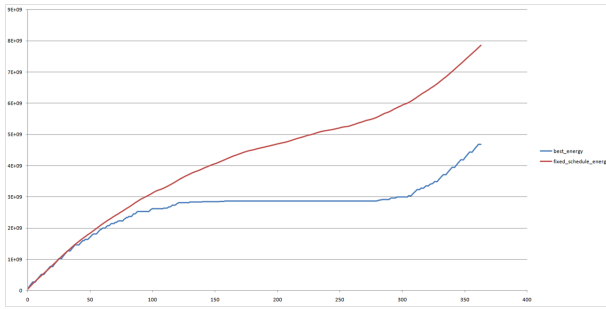


Figure 10: Comparison of energy consumed by our control strategy (blue) and a 5-hour fixed schedule strategy (red). The horizontal axis represents days and the vertical axis represents energy in Joule.

5.6 Adding stochastic inputs to the control framework

The preliminary implementation discussed in the previous section did not include stochastic inputs. In fact, both weather and occupancy are assumed to be deterministic processes over a day. In this section, we discuss how we intend to extend this control framework in order to include stochastic inputs.

As explained in section 3.1, given an exogenous model Φ_O with variables α , we first compute the joint probability distribution $Pr(\alpha)$, and then we generate an ensemble of predictions for α through Monte Carlo sampling; the predictions will then be used as exogenous inputs to the Modelica model Φ_P .

In our application domain, the exogenous model Φ_O is a combination of stochastic weather and occupancy. Therefore, each prediction p for Φ_O will contain the trajectories of weather variables, plus the number of occupants in the room at each time t . In order to use predictions p as exogenous inputs to our control framework, we use algorithm 1. This algorithm computes u^* by searching through a search space composed of $9 \times n$ simulations, where n is the number of predictions for α generated through Monte Carlo sampling.

It is significant to note that the simulated room temperature can change significantly on the basis of different predictions for α . Figure 11 shows an ensemble of 5 room temperature trajectories, obtained with 5 different members of a weather forecast ensemble. Given this significant variability, we believe that using a combination of stochastic weather and occupancy prediction could yield better results than using deterministic forecasts.

Algorithm 1 Algorithm to compute a control action based on an ensemble of exogenous predictions.

for u between 0 and 8 **do**

for each prediction p **do**

calculate $error_u^p$

calculate $energy_u^p$

$J_u^p \leftarrow w_c error_u^p + w_e energy_u^p$

end for

$J_u \leftarrow \sum_{p=1}^n J_u^p$

end for

return u^* for which J_u is minimized

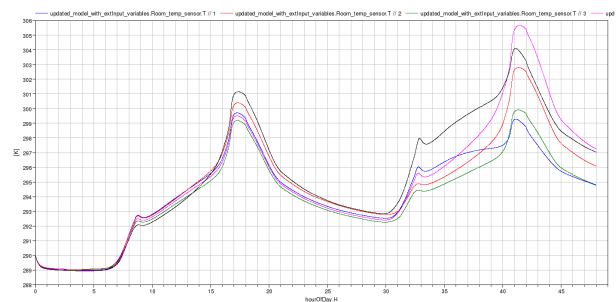


Figure 11: Ensemble of 5 room temperature trajectories, generated with 5 members of a weather forecast ensemble. The horizontal axis contains hours and the vertical axis contains degrees Kelvin.

6 Discussion

We have described an approach to extend Modelica simulation with multiple ensembles generated by exogenous stochastic simulations. This approach couples a discrete-time stochastic simulation with a Modelica simulation, in which the stochastic simulation generates an input to the Modelica model for each time step. Further, the system state from the Modelica model for time t must be used to initialise the model at time $t + 1$. This methodology can enable Modelica to be used for optimisation, and for embedded control and optimisation applications.

Although this approach works well for slower systems, for fast systems (where each time step is small) the computational overhead of initialising a Modelica simulation for each time step hinders real-time and embedded applications. This exposes the limitation of Modelica in two ways: (1) the lack of an in-built stochastic modeling capability; and (2) the inability to accept inputs (e.g., from sensors and actuators) during a simulation. We argue that, in order to gain acceptance for real-world applications, Modelica must ex-

tend its language and computational tools to incorporate methods for dealing with these two deficiencies. Bouskela et al. [1] propose a language extension to partially deal with the first deficiency, but further work is necessary.

References

- [1] Daniel Bouskela, Audrey Jardin, Zakia Benjelloun-Touimi, Peter Aronsson, and Peter Fritzon. Modelling of uncertainties with Modelica. In *Proceedings of the 8th International Modelica Conference*, Dresden, Germany, 2011. Linköping University Electronic Press.
- [2] R. Buizza, P. L. Houtekamer, Gerald Pellerin, Zoltan Toth, Yuejian Zhu, and Mozheng Wei. A comparison of the ecmwf, msc, and ncep global ensemble prediction systems. *Monthly Weather Review*, Vol. 133, No. 5, 2005.
- [3] M. Eames, T. Kershaw, and D. Coley. A comparison of future weather created from morphed observed weather and created by a weather generator. *Building and Environment*, 2012.
- [4] H.R. Glahn and D.A. Lowry. The use of model output statistics (mos) in objective weather forecasting. *Journal of Applied Meteorology*, 11(8):1203–1211, 1972.
- [5] T. Gneiting, F. Balabdaoui, and A.E. Raftery. Probabilistic forecasts, calibration and sharpness. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 69(2):243–268, 2007.
- [6] T. Gneiting and A.E. Raftery. Strictly proper scoring rules, prediction, and estimation. *Journal of the American Statistical Association*, 102(477):359–378, 2007.
- [7] J.A. Hoeting, D. Madigan, A.E. Raftery, and C.T. Volinsky. Bayesian model averaging: a tutorial. *Statistical science*, pages 382–401, 1999.
- [8] C.J. Hopfe and J.L.M. Hensen. Uncertainty analysis in building performance simulation for design support. *Energy and Buildings*, 2011.
- [9] D. Jacob, S. Burhenne, A.R. Florita, and G.P. Henze. Optimizing building energy simulation models in the face of uncertainty, 2010.
- [10] Y. Jiang and T. Hong. Stochastic analysis of building thermal processes. *Building and Environment*, 28(4):509–518, 1993.
- [11] B. Johansson and P. Krus. Probabilistic analysis and design optimization of modelica models. In *Paper presented at the 4th International Modelica Conference*, 2005.
- [12] I.A. Macdonald. *Quantifying the effects of uncertainty in building simulation*. PhD thesis, Department of Mechanical Engineering, University of Strathclyde, 2002.
- [13] M. Wetter. A modelica-based model library for building energy and control systems. In *Proc. of the 11th IBPSA Conference*, 2009.

