

EXPLOITING VARIABLE-STRUCTURE MODELS IN THE CONTEXT OF BUILDING SIMULATIONS WITHIN MODELICA

Jens Möckel¹, Alexandra Mehlhase² and Christoph Nytsch-Geusen¹

¹University of Arts Berlin, Germany

²Technische Universität Berlin, Germany

ABSTRACT

In modeling and simulation of building systems compromises between accuracy and simulation time are often necessary. For performance reasons one would like to use 1D models, but for the sake of accuracy 3D models would be preferred. The 3D models usually have an unfeasible long simulation time. In this paper we propose using variable-structure models in building simulations and switch from 1D to 3D models during simulation depending on the currently necessary accuracy. We will present how such models can be modeled and simulated with currently available models. In a case study of an atrium model we will show that we were able to save up to 70% of simulation time without significant loss of accuracy.

INTRODUCTION

Models in the domain of building systems are often used to study behavior with respect to e.g. technical, cooling, heating, ventilation and economical aspects. Typically, the necessary time invariant models are expressed by differential algebraic equations which can be simulated with suitable solvers.

In general, three different types of modes are used for building and district simulations:

- **0D building models:** Strongly simplified low-order models, modelling building groups up to whole districts.
- **1D building models:** More detailed models of a single building or some of its components, which are normally separated into several thermal zones and where building elements are discretized along one dimension.
- **3D building models:** Very detailed three-dimensional models of the air volumes and their surrounding building elements of one building or a single zone.

These three types of models are referred to as levels of detail.

Complexity of these models has been increased drastically over the last years, even more than the new computer systems. Since in current simulation environments the level of detail of a model is always the same during simulation, the desired accuracy is usually only achievable with unfeasible long simulation times. In building simulation, it is often

not necessary to use the same level of detail during the whole simulation. It would be beneficial to use different models with different levels of detail during one simulation, so the model is always as simple as possible without loss of accuracy. Variable-structure models consist of different modes, where each is represented by a separate model. During a simulation one mode is always active and the simulation can switch from one mode to another one through a transition.

The simulation can therefore switch its level of detail depending on the current circumstances. The advantages are small models, which are easy to maintain and increased portability. The complex behavior of a model can only be executed when needed.

In building simulations such variable-structure models allow for a change of discretization during simulation. For instance, to simulate a room as a 3D model when windows are open or the temperatures need to be accurate. When the room temperature is more or less steady and the windows are closed it might be enough to use a 1D model. Such a mode switch can decrease the needed simulation time while keeping the necessary accuracy.

But it is not only possible to change the level of detail, but also to add or delete components in a model, e.g. opening or closing windows or adding shading elements. It is possible to model such behavior using discretization control methods and event-based mechanisms. But this increases the complexity of the models dramatically and might lead to models which cannot be simulated anymore.

One main advantage of variable-structure modeling over other methods as co-simulation is, that this modeling approach is not closed, but could be later extended by new models, which differ from the already used model in one or more components.

Thus, we want to create a variable-structure model that changes its system of equations to decrease the complexity and therefore the simulation time. Such variable-structure models can already be simulated in specialized frameworks and languages like SOL (Zimmer, 2010), MOSILAB (Nytsch-Geusen C. et al., 2005). In this framework it is for now not possible to use models already existing in Dymola or Simulink. Since our building models are based on the

Modelica library BuildingSystems (Nytsch-Geusen et al, 2013) we want to use common simulation environments for our simulation. The DySMo framework enables such a simulation with common simulation environments. The framework as described in (Mehlhasse, 2014) can control different simulation environments like Dymola, OpenModelica and MATLAB/Simulink. Each mode of a variable-structure model can be implemented in either tool and be simulated in it. The framework handles the switching procedure from one mode to the next. The modeler therefore has to create the modes in a simulation environment and specify the transition between these modes in the framework. To specify the transition a switch condition and an initialization routine for the next mode needs to be defined. The simulation is then controlled by the framework and thus lets the modeler experiment with variable-structure models easily.

In this paper we present how variable-structure models can be advantageously applied to simulating building systems.

We first introduce how modes can be used in building system simulations. We will then discuss how transitions need to be defined when regarding switches between 3D and 1D models. Afterwards we introduce an example of an atrium where we use three different modes and switch between them dependent on the current state of the simulation. We discuss the challenges in modeling such a model and also the advantages for building simulations when using them instead of a conventional approach. Finally, we summarize the main points of this paper and give an outlook.

VARIABLE-STRUCTURE MODELING

In this section we consider variable-structure modelling with respect to the domain of building simulations.

Choosing the modes

Generally speaking, a model consists of a set of equations which describe components and behavior, e.g. physical laws. Therefore, two models can be distinguished, by

1. different sets of equations, i.e. a different set of components and/or different modeled behavior,
2. different type of discretization with respect to space and time or
3. a combination of 1 and 2.

Note that a type of discretization could refer to different numerical methods as well as the same numerical method with different resolution. Note furthermore, that changing the level of detail of a model (0D, 1D, 3D) is not only changing the type of discretization, but also the set of equations with respect to the modeled behavior, i.e. changing the level of detail always refers to 3.

Now concentrating on models in the context of building simulations, we use the term "component" for anything, that is actually part of a building, e.g. the building itself, walls, air volumes, that represents the inner zone of a room or even furniture. With "modeled behavior", we refer to time invariant states, which in most cases have to satisfy conservation equations, e.g. for temperatures, masses and pressures.

We say that a mode is the combination of a model and several simulation settings, e.g. a mode is defined by the model of a room or building and a simulation tool, simulation duration, numerical solver, tolerances and so on.

So we can distinguish between three main groups of modes:

Group1: Modes, that are all based on the same set of simulation settings but differ in used models.

Group2: Modes, that are all based on the same model, but differ in the set of simulation settings.

Group3: Modes which are not covered by Group1 or Group2, i.e. that differ in simulation settings as well as in modeled components.

We consider modes from Group3, i.e. we investigate changes in the actual modeling and the simulation settings. For this group, two modes can always be set in relation to each other either by comparing modeled components, e.g. two different rooms, by the used level of detail, e.g. one 1D and one 3D model of the same room and/or by choosing different simulation settings.

Mode Switches

Switching from one mode to another means to terminate the simulation of a model not only exclusively based on simulation time, but based on simulated variables or events. In most cases, the fulfillment of one or more conditions indicates, that it is necessary to continue the simulation with another model. This might be with respect to the accuracy of results, to simulation time or to a change of simulated behavior, e.g. given the domain of building simulations a previously closed door or window opens.

In order to prevent gaps or jumps in the simulation results, switching from a mode A to a mode B always has the necessity to initialize mode B such that the overall results are continuous. In this case, we speak of a continuous switch. To achieve that, first of all variables have to be identified, which initializes the mode B in a specified state. After that, the representation of each of these variables in mode A has to be determined.

Four cases have to be considered:

Case1: A variable is used across both modes.

Initializing can be realized by setting the

initial value in mode B with the corresponding end value of mode A.

Case2: A variable in mode B corresponds to a number of variables in mode A. In this case, the initial value in mode B can be set by a weighted combination of the end values of the correspondent variables in mode1.

Case3: A number of variables in mode B correspond to one variable in mode A. Initial values in mode B can be set by a weighted fraction of the end value of the correspondent variable in mode A.

Case4: A variable in mode B has no direct correspondence to any variable in mode A. Here, a case by case decision has to be made, e.g. using background knowledge of the model or using variables of related object.

Variables to consider for a continuous switch are mostly physical states, as they will ensure consistency with respect to conservation equations.

This leads to temperatures in each relevant object (e.g. walls, inner zones) and different types of masses in the air (e.g. mass of water in the air, mass of dry air). Most other physical parameters could be derived from that.

As described above, initializing these variables in Case1 and Case4 need no further observation. Given Case2 and Case3 we still have to specify the method for mapping temperatures and masses.

In the following, n gives the number of elements, in which a component is discretized and the superscripts *old* and *new* indicates the mode, a variable is associated with.

For Case2, we simply distribute considered temperatures T to the discretized grid, such that we have a linear mapping:

$$T^{old} \rightarrow T_{[1:n]}^{new} \quad (1)$$

In case of masses m , this mapping has to be adapted, as it would generate mass. Instead we map the n -th part of a considered mass to the finer discretization:

$$\frac{m^{old}}{n} \rightarrow m^{new} \quad (2)$$

For Case3, we differentiate again between temperatures and masses. For temperatures, we compute the mean value of all elements that are condensed, so we have:

$$\frac{\sum_{i=1}^n T_i^{old}}{n} \rightarrow T^{new} \quad (3)$$

For masses we compute the sum of masses of all elements, that are condensed, so we have:

$$\sum_{i=1}^n m_i^{old} = m^{new} \quad (4)$$

Note, that of course different kinds of mappings are possible. The presented mappings (mean value, distribution) have the advantage of simple

implementation but lack in accuracy. Usage of mean values will lead to small gaps in the simulated results, while the distribution of values provide a continuous simulation result, but disregard any additional common knowledge (i.e. height dependencies of temperatures).

But not only the initialization has to be considered but also the guards when a transition needs to be activated. For instance, a mode switch should occur during steady states or when a guard is activated a step back in time might be necessary. In this paper we do not consider these guards in more detail but leave it to future work.

SETUP OF CASE STUDY

Idea for the modes and mode switches

In order to test variable-structure modeling for building simulation, we apply DySMo to a case study of two parts of the building connected to each other by an atrium. The motivation for the case study is to observe the temperature within an atrium and, if the temperature rises up to a predefined condition, to reproduce the natural user-behavior: Cooling the area, e.g. by opening a window.

Both buildings and the atrium are assumed to be shaped as cubicles with an edge length of 10meters. The whole complex is aligned on a west-east axis. Therefore, we refer to the most west building as "Building West", and the most east building as "Building East" respectively. The atrium is simply referred to as "Atrium". Each cubicle is modeled as one zone with six surfaces, which will be referred to as north, east, south, west, ceiling and floor.

For the purpose of thermal masses, Building West and Building East contain walls which divide each cubicle into equally sized rooms.



Figure 1: Basic Setup of Case study

The outer surfaces of the buildings as well as the northern surface and the floor of the atrium are modeled as solid walls. The walls of Building East and Building West contain windows, where the area of windows per wall is half of the walls area. The remaining surfaces of the atrium and the connections between the buildings and the atrium are modeled as walls made of glass. The ceiling as well as the lower part of the south wall of the atrium are moveable, i.e.

it is possible, to open windows for the purpose of cooling by natural ventilation. See Figure 1 for a sketch of this setup.

The above setting is the basic setup. Following the motivation for our case study, we adapt this setup into three different modes:

Mode1: Each zone, each wall and each window is modeled as a 1D object, meaning without any further discretization and without any flows. Physical states are computed as a mean value at the middle of the object.

Mode2: Same parameterization as in Mode1, but the moveable windows are now assumed to be tilted with an angle of 30 degrees.

Mode3: The atriums inner zones as well as the adjacent walls are modeled as 3D objects. All other objects are modeled as 1D objects.

Given the motivation of our case study, these three modes should be used to model following scenario:

Let T_{obs} be the temperature that indicates a possible risk of heat stress, while temperature T_{crit} be the temperature, which is the actual condition for heat stress, called critical comfort condition. Naturally we have $T_{obs} < T_{crit}$. We define T_{atr} as the temperature of the inner air volume of the atrium, which is the state of interest.

If T_{atr} raises to T_{obs} , we want to get detailed data regarding the temperature not only for the whole air volume, but for the region, which is used by human beings, i.e. the bottom part of the atrium up to a height of about. 2.5 meter. Speaking in terms of modeling this means, that the level of detail for the atriums air volume changes.

We now consider the temperature in this bottom part of the air volume. If the temperature decreases the risk of heat stress no longer exists. Therefore, it is rational to reduce the level of detail again. But if the considered temperature increases even more and exceeds T_{crit} , natural reaction of humans will be cooling the area. We realized this behavior by tilting the moveable windows.

While cooling the zone, we also reduce the level of detail with respect to computational time. As soon as temperature in this whole zone falls again below T_{obs} , cooling the area was successful and comes to an end, i.e. the windows are no longer tilted.

Formalized, we have the following conditions to end a mode and initialize a switch to another mode:

Switch1: In Mode1, if $T_{atr} > T_{obs}$, switch to Mode3.

Switch2: In Mode2, if $T_{atr} < T_{obs}$, switch to Mode1.

Switch3: In Mode3,

$$\rightarrow \text{if } \frac{\sum_{i,j=1}^n T_{(i,j,1)}^{atr}}{n^2} > T_{crit}, \text{ switch to Mode2,}$$

$$\rightarrow \text{if } \frac{\sum_{i,j=1}^n T_{(i,j,1)}^{atr}}{n^2} < T_{obs}, \text{ switch to Mode1.}$$

See Figure 2 for a visualization of this scheme. In order to prevent chattering, we set as an additional condition for the activation of a switch, that the model has to be simulated at least some time t_{min} . Through this assumption we do not need to add hysteresis to our conditions.

Note, that in Switch3, n is the order of the 3D discretization of the atrium, assuming the atriums air volume is discretized with an equidistant grid, i.e. into n^3 elements. Temperature of one element is given by $T_{(i,j,k)}^{atr}$, where (i,j,k) gives the coordinates of the discretized box with $(1,1,1)$ is set to be the element in the corner enclosed by wall south, wall to the building west of the atrium and floor. Therefore $\sum_{i,j=1}^n T_{(i,j,1)}^{atr}$ gives the sum of the temperatures of each air volume of the lowermost layer, i.e. the layer closest to the floor. Its division by n^2 leads to the mean value of temperatures closest to the floor.

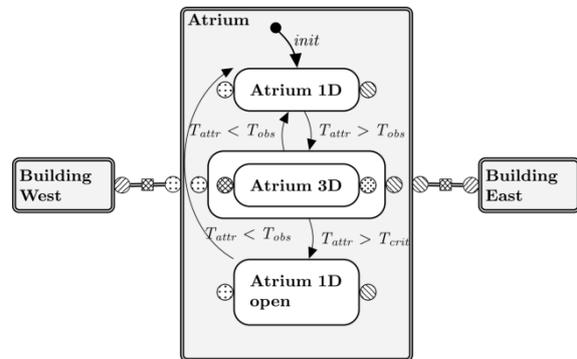


Figure 2: Visualized Switches

Methods of Implementation

This building has been modeled within Dymola 2014 FD1 using objects from the Modelica library BuildingSystems¹.

For the 1D modeling of the inner zones and the walls as well as for the 3D modeling of the walls we use standard elements of this library, i.e. *ZoneTemplateAirvolumeMixed*, *WallThermal3D* or *WallThermal1DNodes*. Note, that even in the case of Mode1 and Mode2 (plain 1D modeling) we use the object *WallThermal3D* with an 1x1 discretization for the northern wall and the floor of the atrium in order to keep this part of the building generic. For the remaining walls, which are assumed to be made of glass, we used the object *Window* with a frame portion near to zero.

For the 3D modeling of the atriums inner zone we use the implementation presented in (Mucha, Nytsch-Geusen, & Streuling, 2014). This approach based on the separation of air volumes and flow elements and is overall an adaption of the finite volume method.

¹See(Nytsch-Geusen, Huber, Ljubijankic, & Rädler, 2013) and (Nytsch-Geusen, Mucha, Inderfurth, & Rädler, 2014) for more information regarding this library.

The inner zone is divided into 64 equally sized air volumes ($4 \times 4 \times 4$), i.e. each volume has an edge length of 2.5 meters. Even if this discretization is still very coarse, it is sufficient enough to observe switching from a 1D model to a 3D model and at the same time follow the motivation of the case study.

In the case of Mode3, we have to consider the connection of the buildings, which are still modeled as 1D objects and the atrium, which is a 3D object. We implemented an adapter-like window. This object transforms $n * m$ surfaces (windows) into one $n \times m$ discretized surface (window). This adapter is also used for the connection of the atrium to the ambient.

Simulation

All tests were performed using a Windows 7 64bit system, with 16 GB Ram and Intel(R) Core(TM) i5-4570 CPU 3.20 GHz. DySMo has been executed with Python 2.7.6². With respect to measuring errors, all results regarding computational times are rounded to full seconds.

For the numerical experiments, we compare several simulations in order to quantify and qualify the effects of structural dynamics. To simplify notation, we use:

Sim1D: Only Mode1 is used (1D modeling).

Sim3D: Only Mode3 is used (3D modeling)

SimSD1: Exploiting structural dynamics, using Mode1 and Mode3 with Switch1 and reduced Switch3 (only second condition is applied). T_{obs} is set to 27°C.

SimSD2: Exploiting structural dynamics, using Mode1, Mode2 and Mode3 with Switch1, Switch2 and Switch3. T_{obs} is set to 25°C, T_{crit} to 27°C.

Table 1 Mode Settings and Properties

| | Mode1 | Mode2 | Mode3 |
|--------------------------|--------------|--------------|------------------------------|
| Solver: | Cvode | Cvode | Cvode |
| Solver Tolerance: | 1e-05 | 1e-02 | 3e-02(Sim3D), 1e-02(else) |
| $t_{min}[s]$: | 600 | 1800 | 600 |
| #continuous time states: | 52 scalars | 53 scalars | 410 scalars |
| # time varying variables | 1006 scalars | 1109 scalars | 6233 scalars |

Table 2 Comparison CPU-Times of Sim1D and Sim3D

| Jan. 1st, 00:00 to Apr., 9th., 00:00 | Sim1D | Sim3D |
|--------------------------------------|-------|----------------------------|
| CPU-Time in [s]: | 21 | 502980 (5d, 19h, 43min) |

See Table 1 for an overview of mode specific simulation parameters and properties. We use Dymola solver Cvode for all modes based on preceding studies. All results are hourly resolved.

²More information about Python can be found here: <http://www.python.org>

To initialize the open window in Mode2, which will be a new object, we use the mapped values of the corresponding closed window of the finished Mode3. Finally, each mode is driven by an ambient, which is based on generated weather data from Berlin.

As a main motivation for structural dynamics we pointed out the unfeasible long simulation times in the case of high resolved results. Table 2 shows a comparison of Sim1D and Sim3D, where Sim3D is nearly 24,000 times slower than Sim1D. It is obvious that there is no actual use case in which the 3D model could be used as a standalone solution. Note, that this Result is based on a simulation of the first three months of a year. Difference in simulation times is likely to increase if it comes to the "hot" summer months, in which larger gradients of state variables (e.g. air temperatures) are present.

To compare Sim1D and Sim3D with SimSD1 and SimSD2, we use results of a two-week simulation (i.e. 1209600 seconds) while a heat stress period, namely July, 17th to July, 31st.

At first, we consider results from Sim1D, Sim3D and SimSD1 (Table 3 and Figure 4). The simulation successfully switched between two different modes several times. Not surprisingly, most of the simulation time is used to simulate, while the simulated time is nearly equally distributed to Mode1 and Mode3. The iterations in the table present how often each mode was active.

Table 3 Statistics SimSD1 for a two-week Simulation

| | Mode1 | Mode3 | Overall |
|------------------------|--------|--------|---------|
| CPU-Time in [s]: | 156 | 13106 | 13262 |
| Simulated Time in [s]: | 637980 | 571620 | 1209600 |
| #Iterations: | 15 | 14 | 29 |

Table 4 CPU-Times for a two-week Simulation

| | Sim1D | Sim3D | SimSD1 |
|------------------|--------|--------|--------|
| CPU-Time in [s]: | 12 | 249101 | 13262 |
| Rel. CPU-Times: | 0.005% | 100% | 5.32% |

In comparison, the simulation time of SimSD1 is just about 5% (Table 4) of the one of Sim3D, while the results are much closer to the high resolved results of Sim3D than the one of Sim1D. The solvers tolerance for Mode3 in SimSD1 is smaller than the one for Mode3 in Sim3D (Table 4 and Figure 3). Furthermore, note that the temperatures for Mode3 are mean temperatures of the discretized cells, which are closest to the floor.

We now consider results from SimSD2 with three modes. Analogue to the presented results for SimSD1 Table 5 shows the CPU time necessary for the simulation, the simulated time and how often each

mode was active. Figure 6 shows the computation time (CPU-time).

Figure 5 shows the mean temperature of the atrium (again in the case of Mode3: Mean temperature of cells closest to floor). As can be seen, the simulated temperature corresponds to the expected behavior for a heat stress period: Temperature rises while the ambient temperature rises (warm air enters the atrium) but cools down much faster during night periods than the simulated temperature in simulation with closed windows.

Table 5 Statistics SimSD2 for a two-week Simulation

| | Mode1 | Mode2 | Mode3 | Overall |
|------------------------|--------|--------|-------|---------|
| CPU-Time in [s]: | 135 | 1323 | 633 | 2091 |
| Simulated Time in [s]: | 547264 | 610583 | 51753 | 1209600 |
| #Iterations: | 14 | 13 | 13 | 40 |

Note, that each time Mode2 is initialized, a peak is simulated. This peak is due to initialization problems based on the transition from an ideal closed system (Mode1, Mode3) to an opened system (Mode2). Therefore, the model needs to balance itself at first. The peaks are compensated after ca. 100 simulated seconds, so we force Mode3 to simulate the corresponding model at least 1,800 seconds, such that this effect does not influence possible transitions. Solving this initialization problem is still work in progress.

In summary, the results show, that applying structural dynamics allows us to combine fast simulations with low-resolved results with slow, but high resolved, simulations. Compared to single high resolved simulations, we reduced the simulation time by 95% and compared to a single low resolved simulation, we get a much more accurate result.

CONCLUSION

In this paper, we have considered variable-structure models within the domain of building simulation using the Python written framework DySMo. An outline of the basic techniques of DySMo has been presented.

We have given an overview of modes and switching modes in general as well as for the domain of building simulations. Additionally, we presented concepts how models should be designed to use them as modes in variable-structure models. Furthermore, we address the problem of initializing a new mode after a mode switch depending on the previous mode and conservation laws.

To evaluate variable-structure models in building systems we created a three zone model with non-trivial wall structure. Of main interest was the atrium, which is one of the zones. This atrium has three modes; 1D, 3D and with open window. To switch

between these modes, we defined transitions defining the switch from one mode to the next depending on the temperature in the atrium. We created two variable-structure models; one which can switch only between 1D and 3D, one which uses all three modes.

We compared the simulation time and simulation results to simulations where only one level of detail was used (3D or 1D). Both variable-structure models were more accurate than using only the 1D model. Furthermore, we were able to reduce the simulation time compared to only using the 3D model up to 70%. This is a great achievement, especially since the loss of accuracy is insignificant in the variable-structure model. Using variable-structure models we are thus able to create models which have a feasible simulation time whilst making sure the simulation results are still valid. We have shown, that the combination of 1D and 3D models enables us to exploit the advantage and avoid the disadvantages of only using one level of detail.

We have shown the application of top level structural dynamics not only for the case of different level of discretization for one model but also for the use case of combining two different set ups. The difference between our two set-ups have been an opened, respectively closed, window. In this case, the simulation results have reproduced the expected temperature curve.

Regarding the problem of initialization, we have shown that the solution for Case1 leads to continuously transitions. The solution for Case2 and Case3 results in small jumps at the time of the actual transitions, which is not surprisingly given the calculated mean value.

Furthermore, results for simulations, that includes Mode2 (building with opened windows), shows the difficulties to identify variables, that are critical with respect to a continuous transition - a difficulty that increases with the complexity of the used model. This corresponds to the usability problems of this method, as this step has to be considered for each mode switch. For now, this means manual work for the modeler. The introduced method is therefore only feasible for models with few modes and switches between modes. The method will be enhanced in the future to allow to create many more and mode switches more easily.

OUTLOOK

For the purpose of a first application of top level variable-structure model to the domain of building system, we have considered a setting of three different modes for a building model and used very simple methods to translate values from one mode to another. Development of variable dependent translation methods is a next step.

Although we consider transitions between different levels of detail as well as transitions between model

with different boundary conditions, the latter one has only been realized by the difference of two opened and closed windows. To push developments in this area more advanced case studies must be implemented.

In our case study, we applied simple termination condition to terminate a mode (comparison of temperatures) and used the final time step as the initial time for the next mode. More evolved methods of termination condition, that include effects of depending variables (i.e. pressure, airflows) as well as further considerations regarding initial time steps will be topics of upcoming research.

One big difficulty has been to identify mandatory states within Dymola, which need to be transferred from one mode to another in order to get consistent results - a work which for the model with opened windows is still in progress. Restricting to physical states has not been purposeful as well as restricting to variables, which are declared as parameters within Dymola. A method to extract essential parameters from a model and the extension of models by information with respect to the application of structural dynamics in order to increase usability is a necessary development.

ACKNOWLEDGEMENT

This study is funded by Deutsche Forschungsgemeinschaft (German Research Foundation) with research grant "Automatische Modellierung und Simulation von technischen Unsicherheiten" (AMSUN), ID Ny 68/5-1.

REFERENCES

- Mehlhase, A. (2014). A Python framework to create and simulate models with ariable structure in common simulation environments. *Mathematical and Computer Modeling of Dynamical Systems, Vol 20*(Iss. 6).
- Mucha, K., Nytsch-Geusen, C., & Streuling, C. (2014). Ein simulationsbasierter Ansatz zur Analyse von Hitzestrebereignissen in Innenräumen. *Proceedings of the 5th German-Austrian IBPSA Conference, 22.-24.September*. Aachen Germany.
- Nytsch-Geusen, C. et al. (2005). MOSILAB: Development of a Modelica based generic simulation tool supporting model structural dynamics. *Proceedings of the 4th International Modelica Conference* (pp. 527-535). Linköping University Electronic Press.
- Nytsch-Geusen, C., Huber, J., Ljubijankic, M., & Rädler, J. (2013). Modelica BuildingSystems – eine Modellbibliothek zur Simulation komplexer energietechnischer Gebäudesysteme. *Bauphysik, Vol. 35*(Iss. 1), 21-29.
- Nytsch-Geusen, C., Mucha, K., Inderfurth, A., & Rädler, J. (2014). Entwicklung eines räumlich und physikalisch adaptiven energetischen Gebäudemodells in Modelica. *Proceedings of the 5th German-Austrian IBPSA Conference, 22.-24.September*. Aachen Germany.
- Zimmer, D. (2010). Equation-based modeling of variable-structure systems. Eidgenössische Technische Hochschule ETH Zürich.

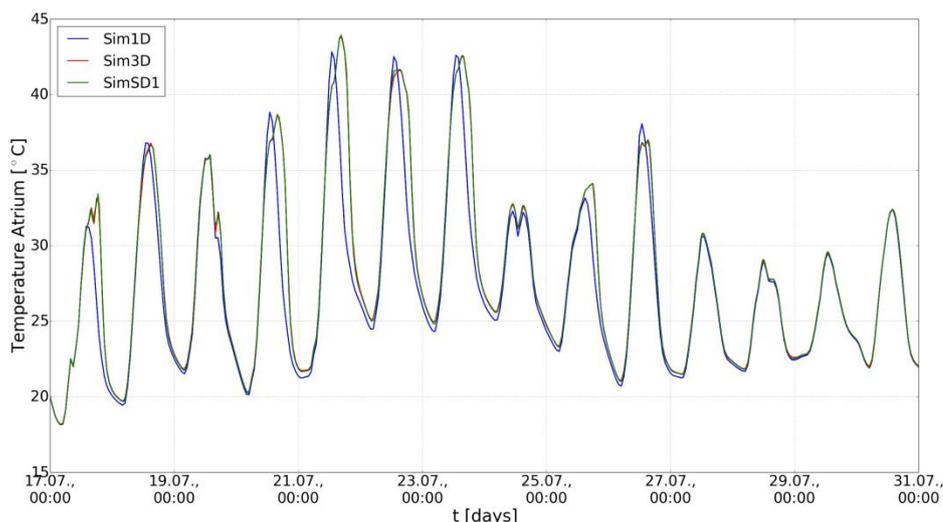


Figure 3 Temperatures SimSD1, Sim1D, Sim3D

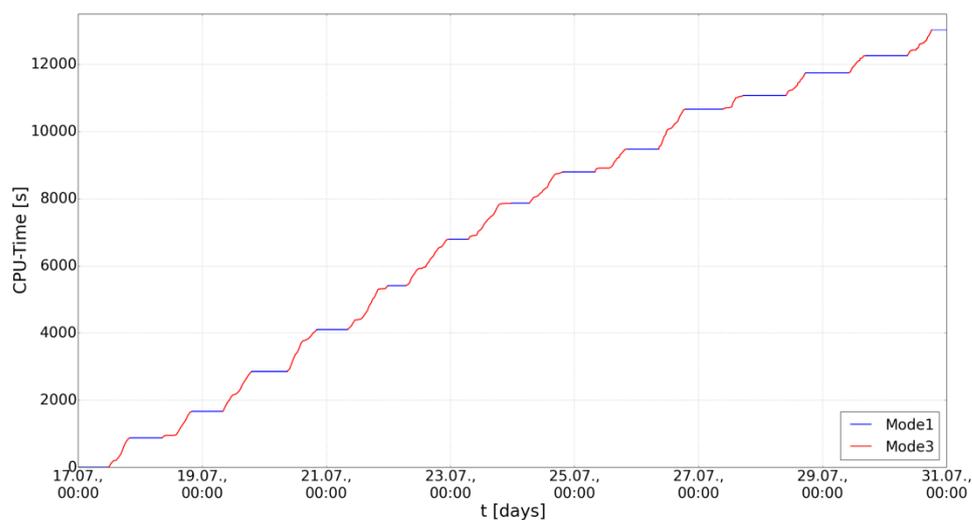


Figure 4 CPU-Time over Simulated Time for SimSD1

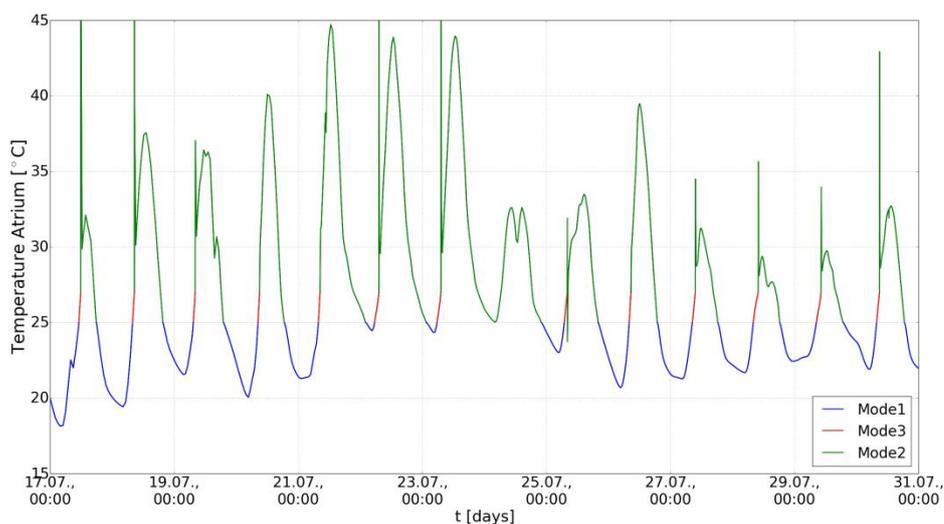


Figure 5 Simulated Temperatures in Atrium over Simulated Time for SimSD2

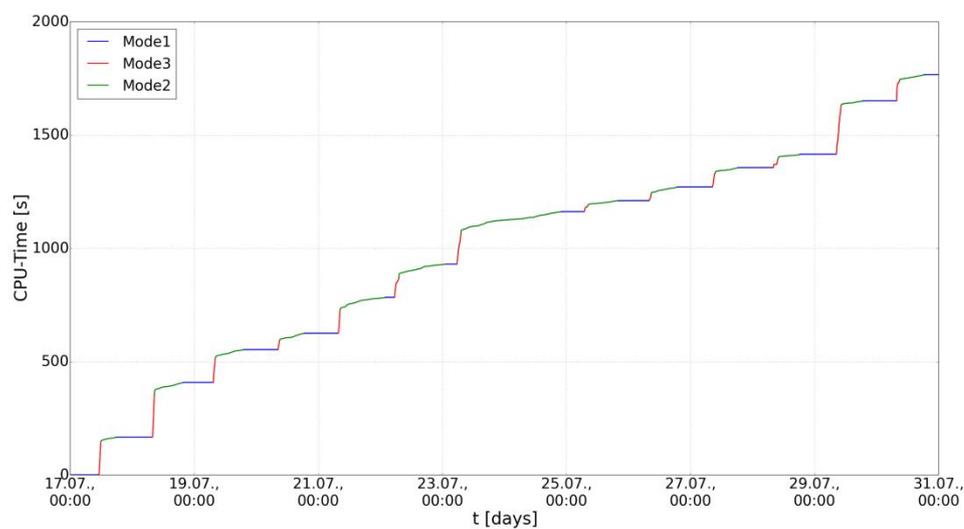


Figure 6 CPU-Time over Simulated Time for SimSD2