SYNERGY OF MATLAB AND MODELICA IN THERMAL FLOWS CONTROL IN BUILDINGS

Borut Zupančič

University of Ljubljana, Faculty of Electrical Engineering
Tržaška 25, 1000 Ljubljana, Slovenia
borut.zupancic@fe.uni-lj.si

ABSTRACT

The paper briefly describes the Modelica model of a cubic shaped room with one window. The 'physical' model was then implemented as a Modelica (Dymola) block in Matlab-Simulink environment. Simulink was used for the realisation of different control schemes, which were 'manually' and 'automatically' optimized. The experiments show that the synergetic combination of Matlab-Simulink and Dymola-Modelica environments is an efficient and powerful approach giving the possibility to realize several important goals: realisation preserving modelling in Modelica, efficient simulation with Simulink and many possibilities for control system design and optimization using basic Matlab and appropriate Matlab toolboxes. However the experiences with Modelica modelling taught us that Modelica models become rather complex and therefore model reduction techniques in order to obtain usable and efficient models are desired. The last part of the paper briefly describes some research activities in this area and also our contributions.

Keywords: OO modelling, control design, model reduction, Modelica

1. INTRODUCTION

Modelling in control is very important in many phases: for the design of new control methods and algorithms, for the implementation of a control algorithm (e.g. model based control systems), for the design of a concrete control system solution but also on higher CIM levels dealing with supervision, fault detection and diagnosis, production supervision, coordination and optimization. The conventional modelling and simulation approach was based on causal block oriented tools, e.g. Matlab-Simulink and before on the so called CSSL languages.

However due to many disadvantages of this approach new modelling techniques were developed in nineties, e.g. Bond graphs and OO acausal and multi domain modelling which preserve the realisation aspects of the systems being modelled. The result was the Modelica language (Fritzson, 2004), (Modelica, 2010) and also the development of several environments (Dymola, Math Modelica, Open Modelica, MapleSim, ...) (Cellier, 1991), (Dymola, 2014).

These modelling techniques were used in our long term activities in modelling of thermal and radiation flows in buildings. We started with this area 15 years ago in cooperation with the Faculty of Civil Engineering, University of Ljubljana. Our first simulator was developed in Matlab-Simulink environment (Škrjanc et al., 2001), (Lah et al., 2005). A miniature test building (cubic shaped, 1m, 1 window) with which we were able to validate the model was also developed. However several disadvantages were noticed: the approach itself was never properly accepted by people from the Civil engineering department, because models in Simulink were difficult to understand. The documentation in Simulink is very problematic and not very transparent. Then we also learned that it is not possible to build the library of reusable components in Simulink. Namely when we wanted to use our one room model for a several rooms model, it was simply not possible. Every new configuration demanded the design almost from the scratch.

Due to these disadvantages we switched to the Dymola-Modelica environment and a new simulator using the standard Modelica library and some components developed by ourselves were developed from the scratch. The results of these developments were published in (Zupančič and Sodja, 2008), (Sodja and Zupančič, 2009), (Zupančič and Sodja, 2013b).

2. MODELLING OF THERMAL AND RADIATION FLOWS IN BUILDINGS IN MODELICA

The basic idea of implementation in Dymola-Modelica is to decompose the system into components that are as simple as possible and then to start from the bottom up, connecting basic components (classes) into more complicated classes, until the top-level model is achieved. The model of the room was built from the prepared model classes. Mostly the model classes from the standard Modelica library for one dimensional thermal processes were used (e.g. `HeatCapacitor`, `ThermalConductor`,

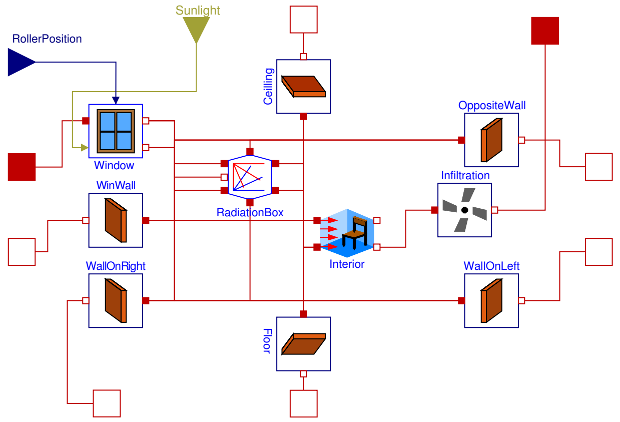

Figure 1: Modelica Model of the Room

`Convection`, `Body Radiation`). The standard connector `Heatport` was also used with heat flow and temperature interface variables.

The appropriate model scheme of the room is shown in Fig. 1. We can notice the model classes for walls, window, floor, ceiling, interior (furniture), etc. All phenomena of thermal flow transfers are taken into account: conduction, convection and radiation. Solar radiation through the window and radiations among the walls and window represent the most complicated modelling parts.

3. PREPARATION OF THE MODELICA MODEL FOR MATLAB-SIMULINK

Dymola-Modelica is an extremely powerful tool for true physical modelling. However for complex experimentations (e.g. optimization, linearization, steady state calculation, etc.), for results presentation it is far from Matlab possibilities. So we decided to use Dymola-Modelica just for the 'physical' part and Matlab-Simulink for all other needs: Simulink for control systems description and Matlab with some toolboxes for making experiments. We prepared a top level Modelica model which can be used as a Dymola (Modelica) block in the Matlab-Simulink environment. Actually we had to prepare appropriate connectors, which are compatible with other Simulink blocks. Such top level Modelica model is shown in Fig. 2. We prepared five inputs (outdoor temperature, roller blind position, direct solar radiation, diffuse solar radiation and artificial heating-cooling) and one output (indoor temperature). Then we prepared Simulink environment to accept Dymola block. This block has to be compiled within Simulink before the simulation is started.

4. CONTOL SYSTEMS OPTIMISATION IN MATLAB

Of course there is no need to use the Matlab environment for pure simulation runs as these can be performed efficiently also in Dymola. However Matlab is efficient if

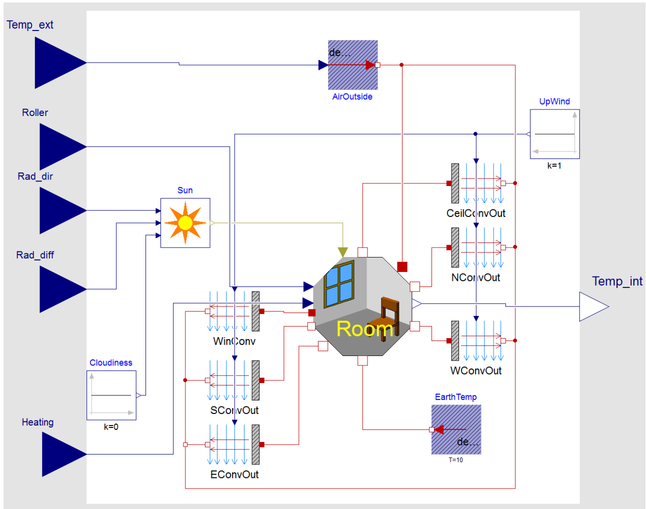

Figure 2: Top Level Modelica Model Intended for the Use within Simulink

we programe more sophisticated experiments using Toolboxes. In the design of control systems we can determine the PID contoller parameters K_P, T_I, T_D

$$u(t) = K_P \left(e(t) + \frac{1}{T_I} \int e(t)dt + T_D \frac{de(t)}{dt} \right) \tag{1}$$

$u(t)$ is the control variable and $e(t)$ is the error or the difference between reference and actual room temperature. Optimisation scheme is shown in Fig. 3. The main Matlab programme actually after initialization calls the optimization function which is supplied also with the special Matlab function for criterion evaluation. Criterion function is evaluated by the help of control systems simulation using Simulink-Dymola model.

Optimization toolbox and unconstrained optimization with the function `fminsearch` were used.

5. EXPERIMENTS WITH P AND PI CONTROLLERS

Although the basic goal was to harmonise the thermal and also radiation flows which influence temperatures and illuminations, we started with more basic experiments to control the internal temperature with additional heating/cooling. Fig. 4 shows the appropriate Simulink diagram. The controller minimises the error between the desired and the actual room temperature. Prior to that, we also performed a number of open loop experiments (Zupančič and Sodja, 2013a). We used a variety of test signals: constants, the step changes as well as signals derived from actual measurements on the test room. Beside usual controller inputs - reference temperature and actual temperature, we added additional input - the signal of direct solar radiation. With this input we intend to improve the control with appropriate feed forward control. The scheme includes the calculation of the criterion functions by means of which an effective manual or auto-

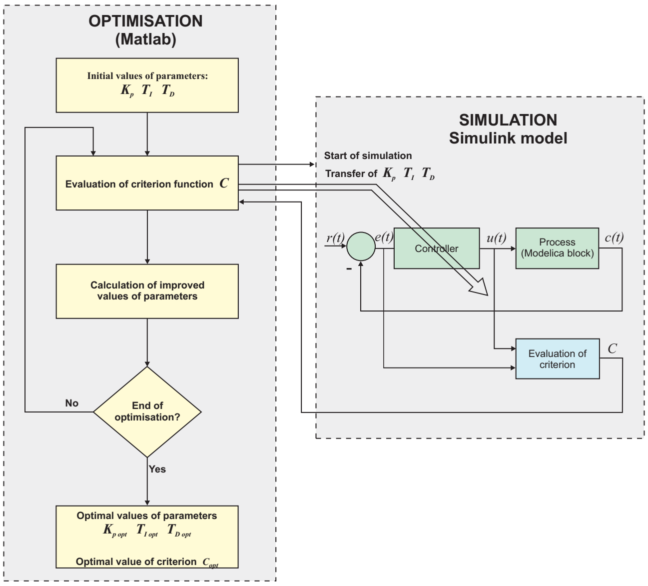

Figure 3: Optimisation Using Matlab, Simulink and Modelica

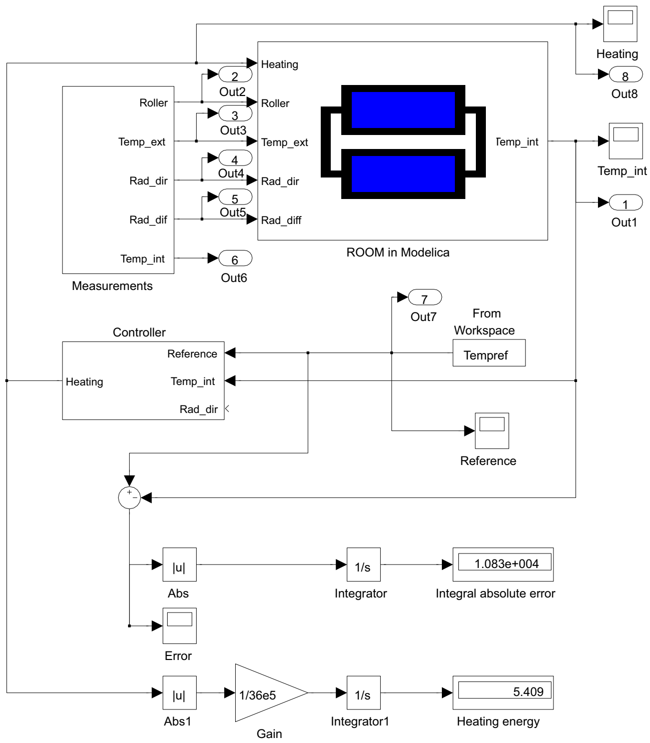

Figure 4: Simulink Scheme with Modelica Model for the Control Experiments

matic tuning (see Fig. 3) is performed. In the first experiment we used a P controller with manually tuned gain $k_P = 50$. With the block `FromWorkspace` the changeable reference room temperature was set. Due to P control a steady state error appeared which was decreased by a higher gain value. The steady state error was also decreased with a superposition of a feedforward signal $20W$ to the control signal.

Using PI (proportional-integral) controller it was possible to significantly decrease the controller gain. The optimisation calculated the gain $k_P = 2$ and the integral time constant $T_I = 10$. Fig. 5 shows the heating/cooling signal and the indoor temperature when the reference temperature changes. The steady state error is small (the biggest value app. 0.6°C). The control signals are also significantly smaller as with P controller (max. 90W). Fig. 6 depicts signals in the environment of real measurements: direct (`Rad_dir`) and diffuse radiation (`Rad_diff`) and outdoor temperatures (`Temp_ext`) were recorded in the period of five days using our pilot set-up. PI controller (with the same parameters as before) was used for the control of indoor temperature (`Temp_int`). In some time instances we also made changes in the roller blind opening (`Roller`). The last two diagrams in Fig. 6 show the heating signal (`Heating`) and the indoor temperature (`Temp_int`). The reference temperature was changed from 15°C to 20°C, 25°C and again to 20°C.

6. REALISATION-PRESERVING MODEL REDUCTION OF MODELS IN MODELICA

Beside described examples we used Modelica with Matlab in many other applications. We learned that OO and multi-domain modelling approach is very efficient especially in model definition phase, but unfortunately not so much in model execution. Namely under the surface of very transparent models very complex structures for execution are obtained. If we use well tested components it does not mean that the model will produce accurate results when many components are put together into a model. If one room model performes accurate results it does not assure that the model with several rooms is also accurate and usable. A simplification and/or model reduction is therefore very important in each modelling application. It is a well-known guideline that a model should not be more complex as necessary for a given purpose. Models satisfying this requirement, i.e. having proper complexity, are often designated as proper models (Wilson and Stein, 1995). However, contemporary component-based modelling approach often yields very detailed models from the beginning and the obtained models can be too complex for many intended tasks. Therefore, automatic model reduction techniques are active research topic and so far numerous automatic model reduction methods have been developed (Ersal, 2007), (Sodja and Zupančič, 2012), (Sodja, 2012). In some fields, e.g., integrated circuits design, they reached a stage when they became an indispensable part of system analysis and hence provided as a part of designated modelling environments (Ugryumova, 2011). The most successful methods, for example, those based on projection techniques, are not realisation-preserving (Ersal, 2007)-the reduced model retains input-output behaviour of the system, but loses physical interpretability of its structure

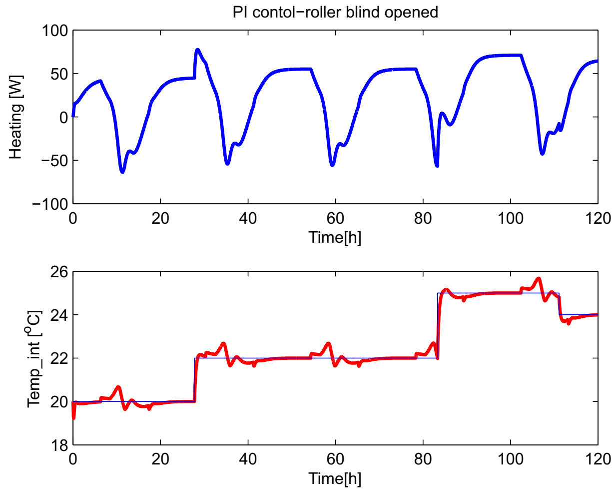

Figure 5: PI Control, Fully Opened Roller Blind

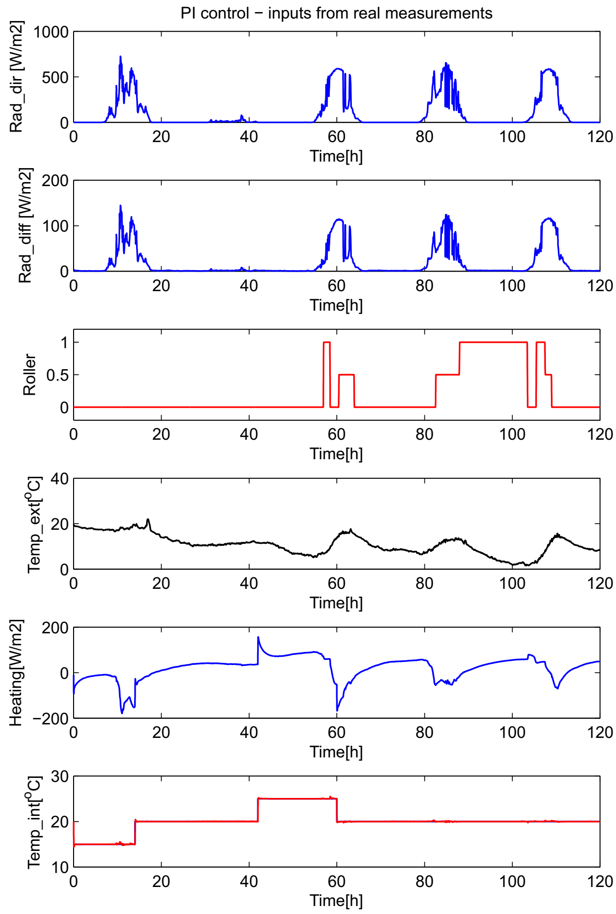

Figure 6: PI Control: Input Signals are Real Measurements

and parameters. In some cases it may be no longer possible to simulate the reduced model with the simulator of the modelling environment which was used at design of the full model. Although preservation of realisation is a very desirable property, realisation-preserving reduction methods are mostly neglected in the literature, mostly due to their bad efficiency. Furthermore, most of existing methods are limited to a certain type of models, e.g., RC circuits (Sheehan, 1999). There are no realisation-preserving model reduction methods known to the author that could adequately handle multi-domain models implemented in contemporary object-oriented modelling languages such as Modelica. Models in Modelica are usually decomposed into several hierarchical levels. At the bottom of the hierarchy, differential-algebraic equations are used for the component description, while on higher levels, model is described by connecting acausal objects (components). This is often done graphically and resulting schematics are called object diagrams (Modelica, 2010). In order to preserve the organisation of original model a combination of model reduction methods is needed. Furthermore, for some tasks, e.g., model verification (Sodja and Zupančič, 2011), only a part of the model might be desired to be reduced.

6.1. Realisation-preserving reduction at object-diagram level

The simplest procedure for reducing models represented with a scheme (graph) is to remove connections (edges) or components (nodes) estimated to have insignificant effect on salient dynamics of the system. Very intuitive approach to determine these connections or components is to use energy and power related metrics. Most energy-related metrics were developed to reduce bond graphs (Louca, 1998), (Ye and Youcef-Youmi, 1999). Bond graphs are object-oriented modelling formalism based on energy and energy exchange and hence very appropriate for energy-based model reduction methods. A power associated with each component is easily obtainable by multiplying variables of the associated bond.

(Louca, 1998) introduced *activity* of elements, an integral of absolute value of all energy the element (submodel) has exchanged with its surroundings within a given time interval $[t_1, t_2]$:

$$\mathrm{A}_i = \int_{t_1}^{t_2} |\sum_j \dot{e}_j(t)| \cdot dt \qquad (2)$$

In Eq. 2 $\dot{e}_j(t)$ designates the j-th energy flow through the boundary of an element. Activity has a physical meaning, it namely represents the amount of energy that flows through the element within a given time interval. It differs from the total RMS energy-flow of an element by putting less weight on the peak values since it uses maximum norm instead of the square averaging which is also in use.

Before element ranking, activities of all elements should be normalised, what means that they are divided by a sum of all elements' activities (total activity of the system)

$$\mathrm{AI}_i = \frac{A_i}{\sum_{j=1}^{n} A_j} \qquad (3)$$

so that a time independent measure is obtained. Normalised activity measure is dubbed *activity index* (AI)

(Louca, 1998).
However, the bond graphs are not the prevalent modelling methodology anymore. Energy, which a component exchanges with its environment, is not so explicitly available in Modelica as in bond-graph formalism (Sodja and Zupančič, 2011), (Sodja and Zupančič, 2012), (Sodja, 2012). However, it can be obtained by inspecting the connections of the components. There are only few different types of physical interactions and therefore types of connections, so if a connector is defined appropriately, a list of rules for calculating power of each connection-type is generated and power associated with a component is calculated as the sum of powers of its connections. Elimination of low ranked components (or connections) in Modelica is even more difficult, because components usually can't be classified in generalised inductance, capacitance and resistance as in case of bond graphs. After ranking of the component is done, it can be whether left to the user to decide how to reduce the model (which is adequate in some cases) or the rules for proper removal of components are derived by automatic manipulation of underlying equations.

6.2. Example: Ranking components in the room model

It was mentioned in (Sodja, 2012) that for each connector of Modelica Standard Library it is possible to determine associated energy-flow considering only information provided by connector's definition. Nevertheless, some connectors are not very appropriately defined for the usage with energy-related metrics. Such an example is the connector for 1-dimensional heat transfer found in library `Modelica.Thermal.HeatTransfer - Interface.HeatPort`. It consists of the effort variable, which is the temperature T, and the flow variable which is the heat-flow rate Q_{flow}. Therefore the energy flow is in this case equal to the flow variable Q_{flow}:

$$\dot{e} = Q_{flow} \tag{4}$$

Namely an extensive (flow) variable of connector for 1-dimensional heat transfer is heat (energy) flow itself, so it disregards the information conveyed by the intensive (potential) variable. The more consistent solution would make us the possibility to obtain heat flow by multiplying the flow and potential variable in the connector. Consider the model presented in section 2. that we developed for the thermal behaviour of the test room. It uses almost exclusively connectors for 1-dimensional heat transfer. In Table 1, components of the room submodel are listed (object diagram is shown in Fig. 1) and sorted according to their *activities*, which were calculated (Eq. 3) for a simulation experiment using measured data for three autumn days. The modelled room has a cubic shape with equal walls, so it was expected that activities of the walls are roughly the same. The results at the bottom of the Table 1 where components `RadiationBox` and `Infiltration` have allegedly zero activity are more surprising. That is because these two components only transfer heat without storing it. Therefore sum of all energy flows on their borders is zero at any time instant. Choice of connector variables where extensive variable is energy flow thus causes that only energy-storing components are considered while transfer-only components are ignored what is by no means acceptable. (Sodja, 2012) used entropy generation rate in Eq. 2 (in place of $\dot{e}$) to evaluate activity metric for a component instead of using heat flow. However the order of components was the same as in Table 1 but with nonzero but still small values of the last two components.

Table 1: Ranking of the Room-Model Components According to the Activity Metric

Element	Activity [J]	Relative [%]	Accumulated [%]
window	$1.38 \cdot 10^7$	22.32	22.325
OppositeWall	$7.76 \cdot 10^6$	12.59	34.91
WinPort	$7.55 \cdot 10^6$	12.25	47.17
WallOppositePort	$5.95 \cdot 10^6$	9.65	56.82
Ceiling	$3.68 \cdot 10^6$	5.97	62.79
WallOnLeft	$3.63 \cdot 10^6$	5.88	68.67
WallOnRight	$3.63 \cdot 10^6$	5.88	74.56
WinWall	$3.44 \cdot 10^6$	5.59	80.14
CeilingPort	$3.23 \cdot 10^6$	5.24	85.39
Floor	$2.77 \cdot 10^6$	4.49	89.88
WallOnRightPort	$1.40 \cdot 10^6$	2.28	92.15
WallOnLeftPort	$1.40 \cdot 10^6$	2.28	94.43
WinWallPort	$1.39 \cdot 10^6$	2.26	96.69
FloorPort	$1.16 \cdot 10^6$	1.88	98.57
Interior	$8.08 \cdot 10^5$	1.31	99.88
OutsideAir	$7.35 \cdot 10^4$	0.12	100.00
RadiationBox	0.01	0.00	100.00
Infiltration	0.00	0.00	100.00

Of course the main question is, what to do with the Table 1. Of course we can not just eliminate the components with low activity, because some classes can not be directly compared. But nevertheless we can find sometimes a very useful information: e.g. the window is very important, the walls have similar importance - perhaps some walls can be modelled with one unified wall etc. Of course if one component between several similar components has much lower activity, we can think how to eliminate this component from the model. In the next section an example will show how the represented ranking can be used for model verification.

Using ranking for model verification

Energy-related metrics were first used for visualisation of dynamic systems modelled with bond graphs (Rosenberg and Ermer, 1995). Analytical models are usually derived from the principle of energy conservation, so it is very intuitive tool for model verification, because it is easy to estimate energy levels of the submodels already in the phase of model design.
Consider a scenario where the component `WallOnLeft`

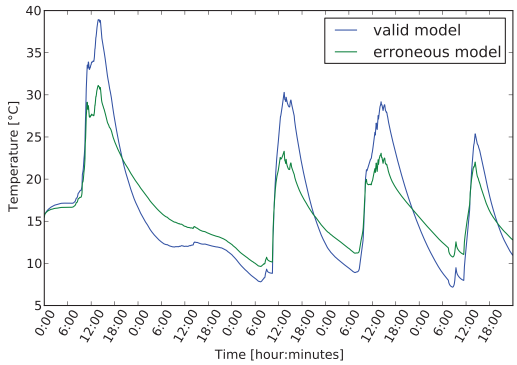

Figure 7: Response of the Erroneous Model Compared to the Response of the Valid Model

Table 2: Component Ranking of the Erroneous Room Model – Component `WallOnLeft` Has the Parameter Brickwork Conductance Set to Ten Times Higher Value.

Element	activity [J]	relative [%]	accumulated [%]
window	$1.34{\cdot}10^7$	21.21	21.21
WallOnLeft	$1.09{\cdot}10^7$	17.25	38.46
WinPort	$7.50{\cdot}10^6$	11.83	50.29
OppositeWall	$6.82{\cdot}10^6$	10.75	61.05
WallOppositePort	$5.91{\cdot}10^6$	9.32	70.37
CeilingPort	$3.14{\cdot}10^6$	4.95	75.32
Ceiling	$3.06{\cdot}10^6$	4.82	80.14
WallOnRight	$2.64{\cdot}10^6$	4.17	84.31
WinWall	$2.39{\cdot}10^6$	3.77	88.08
...	...	...	...

of the model in Fig. 1 has a parameter set to a wrong value. In particular, the conductivity of the brickwork layer is ten times higher as it should be (i.e., it can be caused by a typing error).

Fig. 7 shows the response of this erroneous model in comparison with response of the model having all parameters set correctly. The erroneous model has a distinctly different behaviour than the valid model, so it is easy to detect the presence of error. On the other hand simulation results privide little information to locate the cause of the error. Component rankings, listed in Table 2, is much more elaborate. The component `WallOnLeft` has higher activity as other components also representing walls with same dimension and composition. Therefore, component `WallOnLeft` is probably the cause of error.

6.3. Realisation-preserving reduction at equation level

There are already commercially tools available (Sommer et al., 2008) for reduction and simplification of a general set of differential-algebraic equations. The method combines various algebraic manipulations and approximation techniques, for example, deletion of a single term in an equation, replacement of a term with a constant, deletion of a variable or its derivative, etc. Simplification/reduction operations are ranked according to estimated discrepancies of reduced- and full-model trajectories. The method gives good results for algebraic set of equations, while efficient extension to differential-equation systems is more difficult.

7. CONCLUSIONS

As modelling and simulation is very important in many control design phases it is clear that most recent approaches and tools are desired. In comparison with our former implementation of the model of the room in Matlab-Simulink the OO approach with Modelica significantly improves modelling possibilities. The time for the model development is shortened, the models are more transparent and it is easier for a control engineer to work with area professionals as they better understand Modelica models. In the first part of the contribution we wanted to show the efficacy of this approach modelling a control system for the harmonization of thermal flows in buildings. The combination of Matlab-Simulink and Dymola-Modelica was extremely efficient. Unfortunately with this and even more with some other applications we noticed that such approach has also a limitation due to a huge complexity which appears, when a complex hierarchical structure is flattened for the efficient execution. Therefore some model reduction techniques are even more important in such modelling approaches as in traditional ones. Some methods were tested, developed and also built into a Modelica environment. However all methods are still far from being automatically used. Based on selected metrics a ranking table is obtained. User must carefully analyse the information and try to perform one or more reduction steps. All reductions must be properly verified. By now we actually did not succeed to make some efficient and automatized reductions in our complex applications but we were able to obtain some good results in more simple test examples. So there are many possibilities for the future work: to develop new or improved methods for reduction of object diagrams and equations but also for the implementation of appropriate procedures in modelling compilers.

REFERENCES

Cellier, F. E., 1991. *Continous System Modeling*, Springer Verlag, New York.

Dymola, (2014). *Dymola, Dynamic Modeling Laboratory, User's Manual*, Dassault Systèmes.

Ersal, T., (2007). *Realization-preserving simplification and reduction of dynamic system models at the graph level*, PhD thesis, University of Michigan.

Fritzson, P., 2004. *Principles of Object Oriented Modeling and Simulation with Modelica 2.1*, IEEE Press, John Wiley & Sons, Inc., USA. Publication.

Lah, M. T., Zupančič, B. and Krainer, A., 2005. Fuzzy control for the illumination and temperature comfort in a test chamber, *Building and environment*, 40 (12), 1626–1637.

Louca, L. S., (1998). *An Energy-based Model Reduction Methodology for Automated Modeling*, PhD thesis, University of Michigan.

Modelica, (2010). *Modelica Specification, version 3.2*. `www.modelica.org/documents/ModelicaSpec32.pdf`.

Rosenberg, R. C. and Ermer, G. E., 1995. A bond graph visualization tool to improve engineering system design, *Systems Analysis, Modeling, Simulation*, 18–19, 173–178.

Sheehan, B. N., 1999. TICER: Realizable reduction of extracted RC circuits, *IEEE/ACM International Conference on Computer-Aided Design, Digest of Technical Papers*, pp. 200–203.

Sodja, A., (2012). *Object-oriented modelling and simulation analysis of the automatically translated models*, PhD thesis, University of Ljubljana, Fac. of El. Eng.

Sodja, A. and Zupančič, B., 2009. Modelling thermal processes in buildings using an object-oriented approach and Modelica, *Simulation modelling practice and theory*, 17 (6), 1143–1159.

Sodja, A. and Zupančič, B., 2011. On using model aproximation techniques for better understanding of models implemented in Modelica, *Proceedings of the 8th International Modelica Conference*, Dresden, Germany, pp. 697–703.

Sodja, A. and Zupančič, B., 2012. Realisation-preserving model reduction of models in Modelica, *Proceedings of the 7th Vienna Conference on Mathematical Modelling*, Vienna, Austria, pp. 322–328.

Sommer, R., Halfmann, T. and Broz, J., 2008. Automated behavioral modeling and analytical model-order reduction by application of symbolic circuit analysis for multi-physical systems, *Simulation Modelling Practice and Theory*, 16, 1024–1039.

Ugryumova, M. V., (2011). *Applications of Model Order Reduction for IC Modeling*, PhD thesis, Eindhoven University of Technology.

Škrjanc, I., Zupančič, B., Furlan, B. and Krainer, A., 2001. Theoretical and experimental fuzzy modelling of building thermal dynamic response, *Building and environment*, 36 (9), 1023–1038.

Wilson, B. H. and Stein, J. L., 1995. An algorithm for obaining proper models of distributed and discrete systems, *ASME Journal of Dynamic Systems, Measurement and Control*, 117, 534–540.

Ye, Y. and Youcef-Youmi, K., 1999. Model reduction in the physical domain, *Proceedings of the American Control Conference*, San Diego, CA, USA, pp. 4486–4490.

Zupančič, B. and Sodja, A., 2008. Object oriented modelling of variable envelope properties in buildings, *WSEAS transactions on systems and control*, 3 (12), 1046—1056.

Zupančič, B. and Sodja, A., 2013a. Analysis and control design of thermal flows in buildings: efficient experimentation with a room model in matlab-modelica environment, *Proceedings of the 8th Eurosim Congress on Modelling and Simulation*, Cardiff, UK, pp. 155–160.

Zupančič, B. and Sodja, A., 2013b. Computer-aided physical multi-domain modelling: some experiences from education and industrial applications, *Simulation modelling practice and theory*, 33 (6), 45–67.

AUTHORS BIOGRAPHY

BORUT ZUPANČIČ received his Ph.D. and became a full professor at the Faculty of Electrical Engineering, University of Ljubljana in 2000. His major research interests are: control systems, multi-domain and object oriented modelling and simulation, continuous and hybrid control systems design, harmonization of thermal and flows in buildings. He is the author of more than 200 conference papers and 50 papers in scientific journals, co-author of one international book (published by Prentice Hall Inc.) and author or co-author of several books in Slovene language. He was the president of EUROSIM - the Federation of European Simulation Societies 2004-07 and the president of SLOSIM - the Slovene Society for modelling and simulation. Currently he is the secretary of EUROSIM and the Head of the Laboratory for Modelling, Simulation and Control at the University of Ljubljana, Faculty of Electrical Engineering. His Web page can be found at `http://msc.fe.uni-lj.si/Staff.asp?person=2`.