Computer aided support for the temperature control in buildings

Borut Zupančič

Faculty of Electrical Engineering,
University of Ljubljana,
Tržaška 25, 1000 Ljubljana, Slovenia
Email: 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' optimised. The experiments show that the synergetic combination of MATLAB-Simulink and Dymola-Modelica environments is an efficient and powerful approach giving the possibility to realise several important goals: realisation preserving modelling in Modelica, efficient simulation with Simulink and many possibilities for control system design and optimisation 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: object oriented; OO modelling; multi-domain modelling; thermal flows; radiation flows; temperature control; control design; PID control; optimisation; model reduction; Modelica.

Reference to this paper should be made as follows: Zupančič, B. (2017) 'Computer aided support for the temperature control in buildings', *Int. J. Simulation and Process Modelling*, Vol. 12, No. 6, pp.459–469.

Biographical notes: Borut Zupančič received his PhD 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, harmonisation of thermal and radiation 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 in 2004–2007, the Secretary of EUROSIM in 2010–2016 and the President of SLOSIM – the Slovene Society for Modelling and Simulation in 1994–2002 and 2010–2014. Currently, he is the Head of the Laboratory for Modelling, Simulation and Control at the University of Ljubljana, Faculty of Electrical Engineering.

This paper is a revised and expanded version of a paper entitled 'Synergy of Matlab and Modelica in thermal flows control in buildings' presented at I3M 2015, Bergeggi, Italy, 21–23 September 2015.

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 computer integrated manufacturing (CIM) levels dealing with supervision, fault detection and diagnosis, production supervision, coordination and optimisation. Although the modeller's knowledge and intuition is extremely important, modern tools and environments that support also real time experimentations are urgent (Huang et al., 2015). 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 object oriented (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, ...) (Çellier, 1991).

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, 1 m, 1 window) with which we were able to validate the model was also developed (see Figure 1).

Figure 1 A miniature test building (see online version for colours)

However several disadvantages with MATLAB-Simulink modelling 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.

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

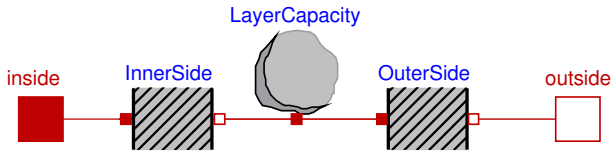
2 Modelling of thermal and radiation flows in buildings in Modelica

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

A wall normally consists of several layers. The resulting Modelica scheme for a one-layer implementation is shown in Figure 2.

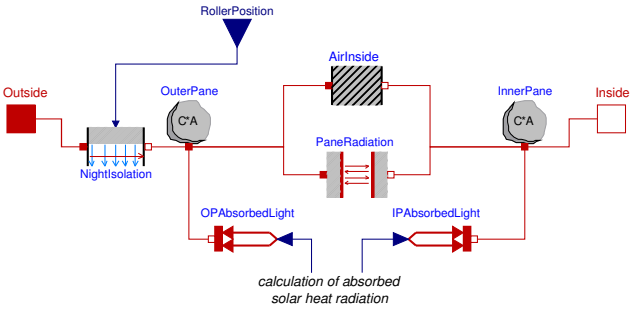
The block called the `LayerCapacity` is a model of a heat capacitor, while the blocks `InnerSide` and `OuterSide` are models of the thermal conduction through the layer, and are connected on one side with the `LayerCapacity` and on the other side with the stand-alone connectors `inside` and `outside`. The described structure is defined as a `Layer` model class. There are three connecting points with three different temperatures: the average temperature in the middle of the layer, and two boundary-layer temperatures on both sides. The model of the wall is obtained by simply connecting several layer sub-models in series. The structure of the wall is further connected to the other connectors according to the wall's boundary conditions.

Figure 2 Scheme of a wall layer in Modelica (see online version for colours)



A similar procedure was used to implement the model class of a window. The scheme is shown in Figure 3.

Figure 3 Scheme of the model describing the thermal processes in a window in Modelica (see online version for colours)



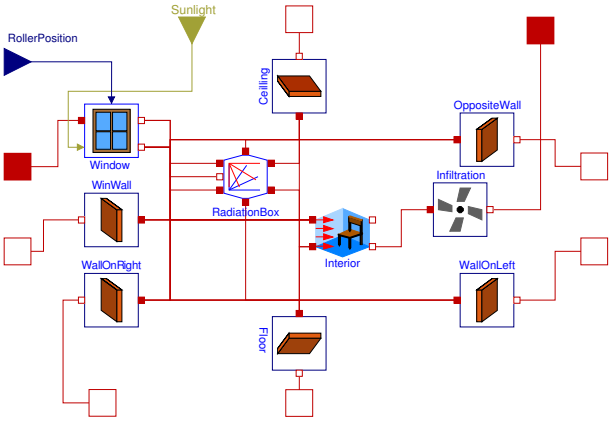
The heat capacities of the outer and inner panes are modelled with two `HeatCapacitor` model classes, `OuterPane` and `InnerPane`, the connectors of which also contain the panes' average temperatures. Both panes interact with each other via thermal radiation and thermal conduction through the air in the gap between the panes. Therefore, `OuterPane` and `InnerPane` are connected with

the model classes `AirInside` and `PaneRadiation`. There are also model classes named `OPAbsorbedLight` and `IPAbsorbedLight` in Figure 3. These are conversion blocks and transform the absorbed solar-radiation flows into connections of the panes' heat-capacity blocks. They are needed to convert the absorbed radiation flows, calculated as a real variable, into the `HeatPort` connector type. A more detailed description of the solar-radiation flows and the appropriate Modelica implementation can be found in Sodja and Zupančič (2009).

All the other blocks, which model other thermal flows coming from the window's surroundings, are connected to the stand-alone connectors `Outside` and `Inside`, respectively. It is clear that the connector `Outside` is not connected directly to the heat-capacity model class `OuterPane` of the pane in Figure 3, but through the `NightIsolation` model class that models the influence of the roller blind to the thermal conductance through the window.

Finally, a model of a room can be built from the prepared model classes. The overall scheme consists of classes that model the room's envelope and those from the interior model class. The appropriate model scheme is shown in Figure 4.

Figure 4 Modelica model of the room (see online version for colours)

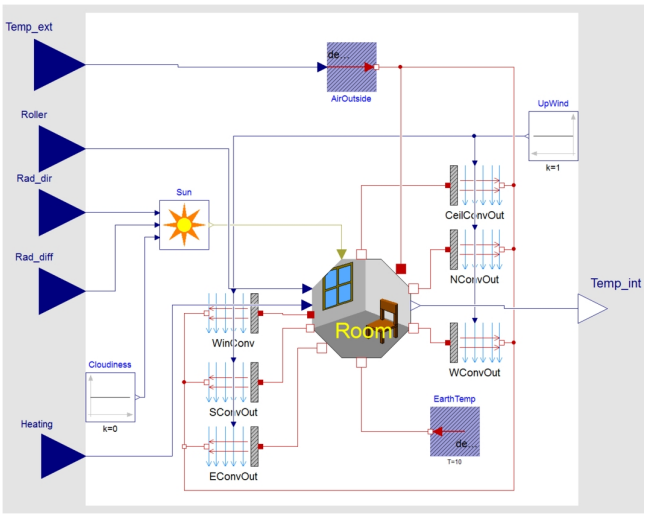


The class *Interior* in the middle is surrounded with the classes of the room envelope. The inner surfaces of the envelope (represented by the connector facing towards `Interior`) are connected to the `RadiationBox` class, which models the thermal radiation exchange between the surfaces (and is beyond the scope of this paper), `Interior` class (model of the air mass and the furniture inside) and to the lower-right connector of the `Window` class, which is an array of solar-radiation heat flows received by each surface. The external surfaces of the envelope are connected to connectors that are visible from the outside of the model of the room, and were used in the top-level model. The blocks that model the convection between the outdoor air and the walls of the building (ceiling, north, south, east and west walls) are therefore connected to those connectors. The `Floor` connector is connected to a constant ground temperature. The intensity of the solar radiation is routed to a class named `Sun` in the top-level model, where the direction vector of the solar rays is also calculated from a specified start date and simulation time and packed together with the solar-radiation component intensity into one connector (`Sunlight` in Figure 4).

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., optimisation, linearisation, 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 Figure 5. 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.

Figure 5 Top level Modelica model intended for the use within Simulink (see online version for colours)



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 we programe more sophisticated experiments

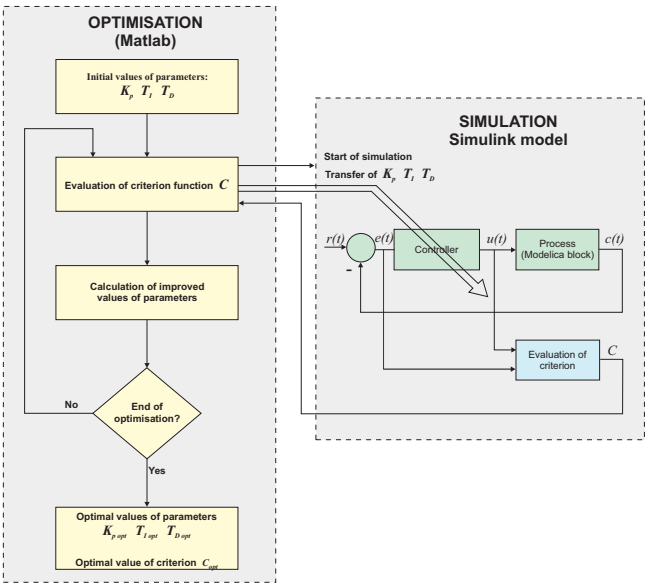
using Toolboxes. In the design of control systems we can determine the PID contoller parameters K_P, T_I, T_D

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

$u(t)$ is the control variable and $e(t)$ is the error or the difference between reference and actual room temperature. K_P is the controller, T_I and T_D are the controller integral and derivative constants.

Optimisation scheme is shown in Figure 6. The main MATLAB program after initialisation calls the optimisation 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.

Figure 6 Optimisation using MATLAB, Simulink and Modelica (see online version for colours)



Optimisation toolbox and unconstrained optimisation with the function `fminsearch` were used.

5 Open and close loop experiments

5.1 Experiments in the open loop

We started the control systems design with a number of open loop experiments. The Simulink scheme with Modelica block is shown in Figure 7.

We used a variety of test signals: constants, the step changes as well as signals derived from actual measurements on the mentioned miniature test room. In the scheme in Figure 7 the roller blind is closed, the heating, the direct and diffuse radiations are constants and the outdoor temperature is defined in MATLAB workspace. Indoor and outdoor temperatures are observed.

Figure 8 shows the indoor temperatures (roller blind closed, halve opened, fully opened) when the 50 W heater was switched on after 60 h. The observation time is 5 days or 120 h.

Figure 7 Simulink scheme with the Modelica model for open loop experiments (see online version for colours)

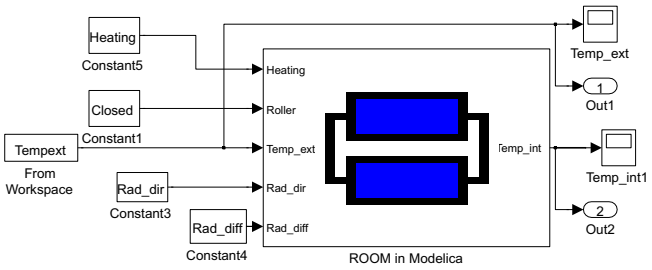
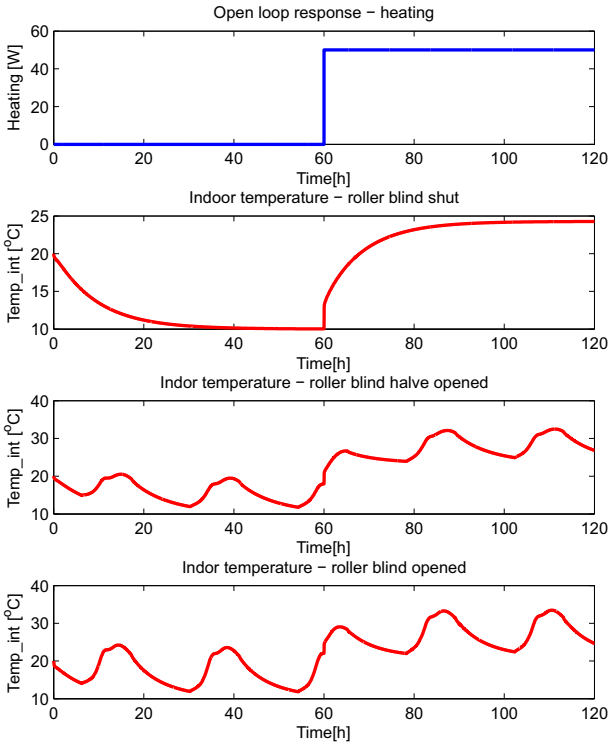


Figure 8 Open loop experiment-step change of heating (see online version for colours)

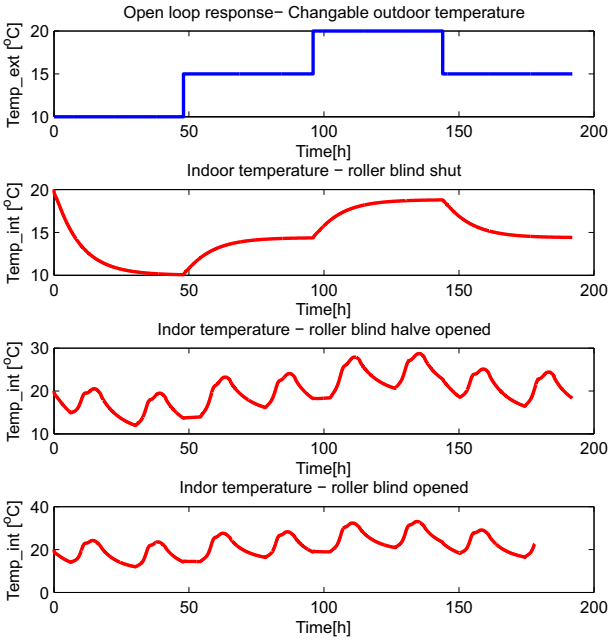


The direct and diffuse radiations had constant values (300 W/m^2 and 50 W/m^2), the outdoor and terrain temperatures had the constant value 10°C. However due to the changeable sun position during the day the amount of radiation passing through the window is also changeable and the influence to the indoor temperature is such that a periodic behaviour is obtained, when the roller blind is not completely closed. When it is half opened, the temperature oscillates for app. 7.5°C, and when it is fully opened for app. 11.5°C. When the window is fully shaded the temperature increases for 15°C (see first response, which also proves the nonlinear behaviour).

Figure 9 shows the indoor temperatures (roller blind shut, halve opened, fully opened) when the outdoor temperature changes from 10°C to 15°C, to 20°C and to 15°C. The observation time is 200 h. There was no heating and the temperature of the terrain was 10°C. The outdoor temperature changes each two days. The steady state of the

indoor temperature is not equal to the outdoor temperature because the temperature of the terrain, which is constant 10°C, also influences the heating process.

Figure 9 Open loop experiment-changeable outdoor temperature (see online version for colours)

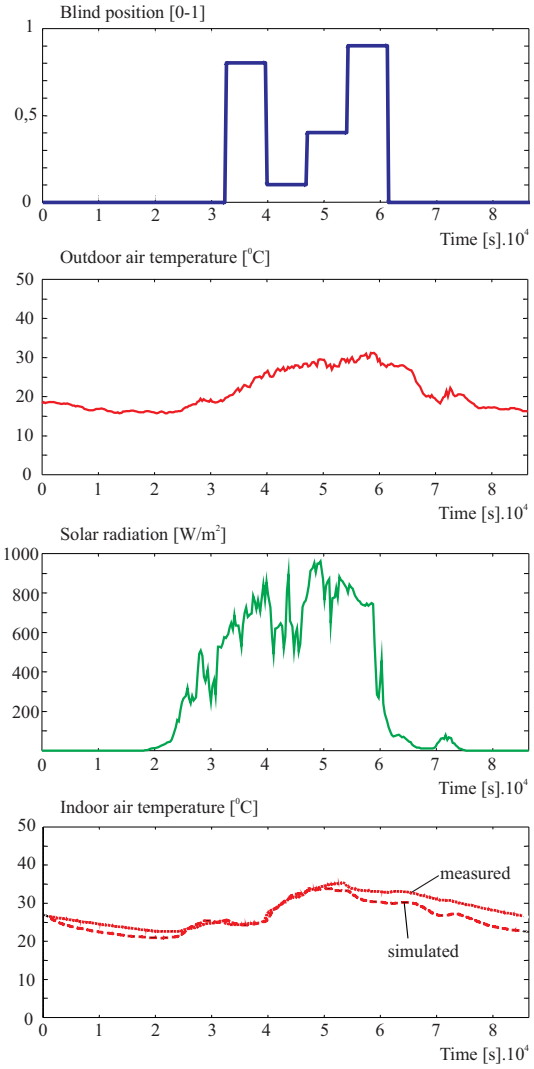


5.2 *Model validation*

Such simple open loop experiments are especially important for model validation as they prove that the model behaves similar to the real system. Model validation is the most important phase in each modelling and simulation iterative cyclic procedure. It is based on the comparison of experimental and simulation results, when the real system and simulation model are influenced by the same input signals. It is of course important to perform several experiments under different conditions. The experimental data should be different from those used in model development phase. The real experiments were performed in winter, spring, summer and autumn conditions, with heating and ventilating and with roller blind positioning. The real system and the simulation model were influenced with several external temperature signals, with solar radiation signals and with some variable properties of the envelope of the room. It is desired that input signals contain appropriate dynamics. With these experiments the model was still improved with final parameter tunings.

Figure 10 shows an experiment from a late spring period (beginning of June). In this experiment the window shading area is also changeable. The observation period is 24 hours, the sampling time is 5 min. The roller blind position (0 – completely opened, 1 – completely shaded), the outdoor temperature (app. 16°C–31°C) and the global solar radiation (max. 950 W/m^2 at 2 pm) are shown in the first three diagrams. The lower diagram depicts the measured and simulated indoor temperatures. The error range is acceptable and is probably caused by non-modelled phenomena such as unexpected ventilation heat losses through some cracks in the dry wall panels and the influence of wind.

Figure 10 Measured and simulated indoor temperatures as a result of variable outdoor temperature, solar radiation and window shading area (see online version for colours)



5.3 *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. Figure 11 shows the appropriate Simulink diagram. The controller minimises the error between the desired and the actual room temperature. 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 automatic tuning (see Figure 6) is performed.

Figure 11 Simulink scheme with Modelica model for the control experiments (see online version for colours)

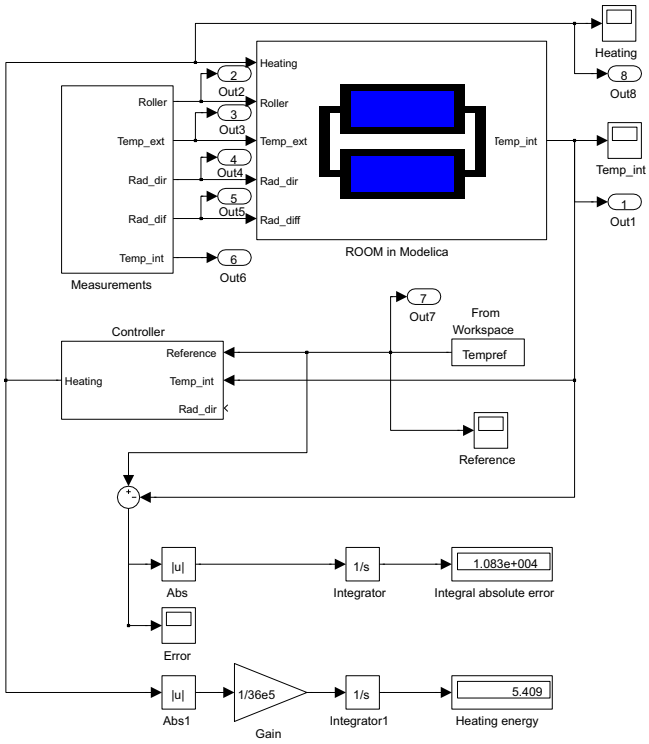
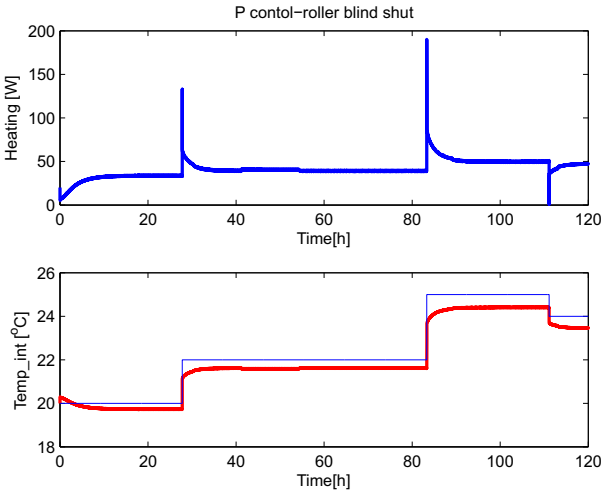


Figure 12 P control, shaded window (see online version for colours)



In the first experiment P (proportional) controller was used with the gain $k_P = 50$. With the FromWorkspace block we defined the changeable reference indoor temperature. As the steady state error was not negligible we introduced a feed forward control signal 20 W, which was added to the control signal. Figure 12 shows the response (heating, reference and indoor temperature) with the fully shaded window (max. error 0.5°C). Figure 13 shows the situation with completely opened roller blind (max. error 1°C). To assure a relatively small steady state error the gain of the controller was rather high, therefore the control signals were big in the moments of the reference changes (up to 150 W).

Figure 13 P control, fully opened roller blind (see online version for colours)

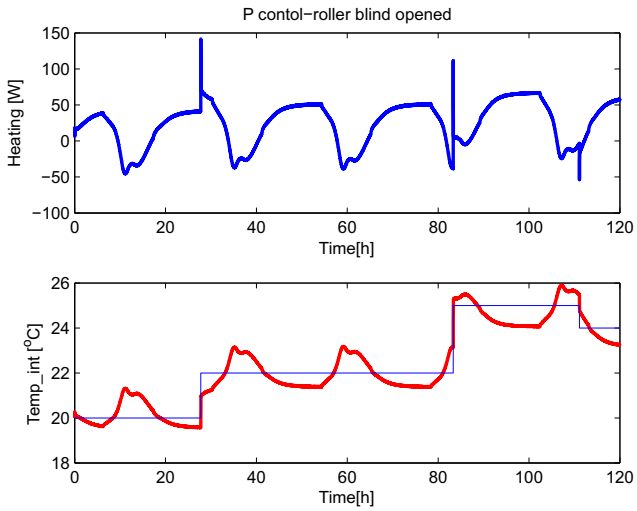
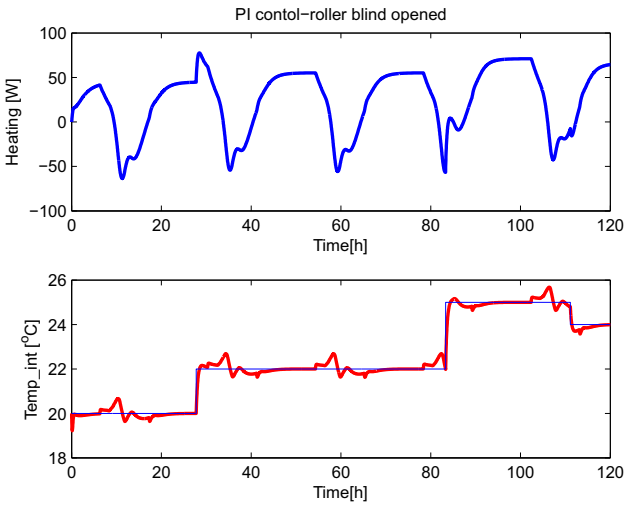


Figure 14 PI control, fully opened roller blind (see online version for colours)



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 $T_I = 10$. Figure 14 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. 90 W).

In the next experiment we tested the ability of the same control system (with the same parameters) for a disturbances elimination. In this case the reference indoor temperature was fixed to 20°C while we simulated the changeable outdoor temperature (each two days the step increased for 5°C). Figure 15 shows the heating and the indoor temperature. We also calculated the energy consumption in 5 days, which was 2.53 kWh. We can also comment that when the outdoor temperature is 20°C, we still need a small amount of heating although the reference temperature is also 20°C, because we use the temperature of the terrain 10°C in simulation studies.

Figure 15 PI disturbance control: roller blind is completely closed (see online version for colours)

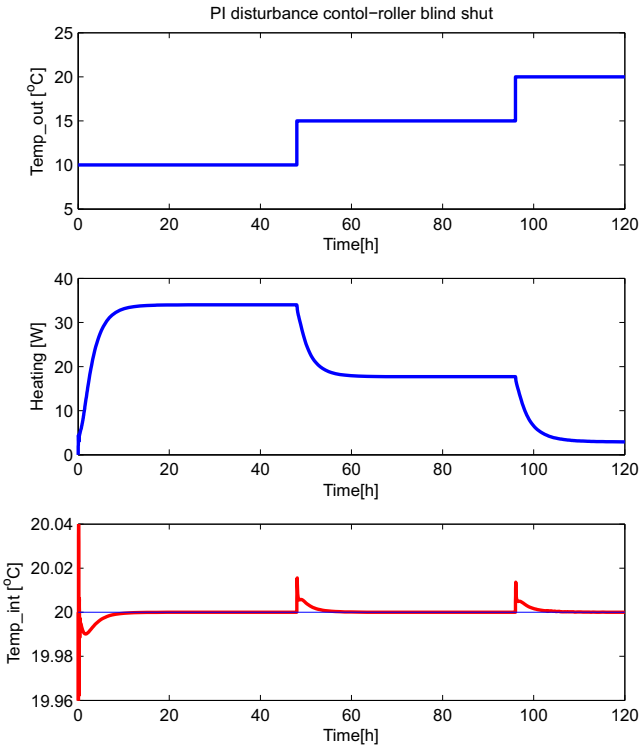
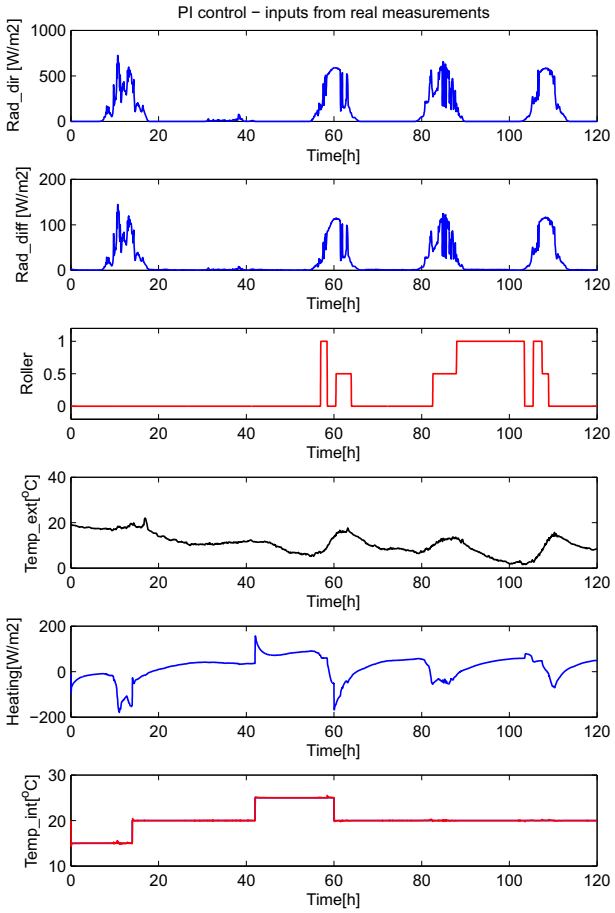


Figure 16 depicts signals in the environment of real measurements: direct (`Rad_dir`) and diffuse radiation (`Rad_diff`) and outdoor temperatures (`Temp_ext`) were recorded in the period of five days using our real miniature room. PI controller (with the same parameters as before) was used for the control of indoor temperature (`Temp_int`). In some time instances we also made changes in the roller blind opening (`Roller`). The last two diagrams in Figure 16 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.

As noted the PID controller parameters were tuned manually and by an optimisation. Of course we see many possibilities for future investigations with other tuning approaches (Wang et al., 2015) but also with the usage of other control algorithms, e.g., predictive control algorithms (Salem et al., 2015).

Figure 16 PI control: input signals are real measurements (see online version for colours)



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 performs 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 than 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 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 $[t_1, t_2]$:

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

In equation 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)

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

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, 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 cannot 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 Figure 4) and sorted according to their *activities*, which were calculated [equation (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 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 equation (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 non-zero but still small values of the last two components.

Of course the main question is, what to do with 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.

6.2.1 *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.

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

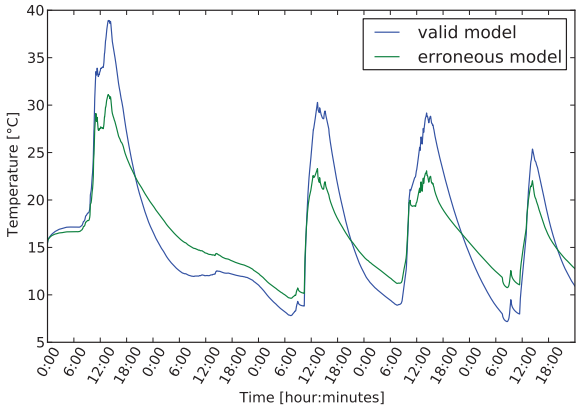
Table 2 Component ranking of the erroneous room model – component `Wallonleft` has the parameter brickwork conductance set to ten times higher value

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
...	...	...	...

Consider a scenario where the component `WallOnLeft` of the model in Figure 4 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).

Figure 17 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 provide 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.

Figure 17 Response of the erroneous model compared to the response of the valid model (see online version for colours)



6.3 Realisation-preserving reduction at equation level

There are already tools commercially 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 thermal and radiation flows in 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 harmonisation 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 automatised 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

Çellier, F.E. (1991) *Continous System Modeling*, Springer Verlag, New York.

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.

Huang, H., Nowoisky, S., Knoblich, R. and Gühmann, C. (2015) 'Multi-domain modelling and simulation of an automated manual transmission system based on Modelica', *Int. J. of Simulation and Process Modelling*, Vol. 10, No. 3, pp.253–264.

Lah, M.T., Zupančič, B. and Krainer, A. (2005) 'Fuzzy control for the illumination and temperature comfort in a test chamber', *Building and environment*, Vol. 40, No. 12, pp.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 [online] http://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*, Vol. 18–19, pp.173–178.

Salem, F., Mosaad, M. and Awadallah, M. (2015) 'A comparative study of MPC and optimised PID control', *Int. J. of Simulation and Process Modelling*, Vol. 2, No. 4, pp.242–250.

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.

Škrjanc, I., Zupančič, B., Furlan, B. and Krainer, A. (2001) 'Theoretical and experimental fuzzy modelling of building thermal dynamic response', *Building and Environment*, Vol. 36, No. 9, pp.1023–1038.

Sodja, A. and Zupančič, B. (2009) 'Modelling thermal processes in buildings using an object-oriented approach and Modelica', *Simulation Modelling Practice and Theory*, Vol. 17, No. 6, pp.1143–1159.

Sodja, A. and Zupančič, B. (2011) 'On using model approximation 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.

Sodja, A. (2012) *Object-oriented Modelling and Simulation Analysis of the Automatically Translated Models*, PhD thesis, University of Ljubljana, Fac. of El. Eng.

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*, Vol. 16, pp.1024–1039.

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

Wang, H., Hu, Y., Liao, W. and Yan, T. (2015) 'Optimal PID control of DC motor with ABC and PSO algorithms', *Int. J. of Simulation and Process Modelling*, Vol. 6, No. 5, pp.193–200.

Wilson, B.H. and Stein, J.L. (1995) 'An algorithm for obtaining proper models of distributed and discrete systems', *ASME Journal of Dynamic Systems, Measurement and Control*, Vol. 117, pp.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*, Vol. 3, No. 12, pp.1046—1056.

Zupančič, B. and Sodja, A. (2013) 'Computer-aided physical multi-domain modelling: some experiences from education and industrial applications', *Simulation modelling practice and theory*, Vol. 33, No. 6, pp.45–67.