

A review of the application of SimuLink S-functions to multi domain modelling and building simulation

A.W.M. (Jos) van Schijndel*

Building and Architecture, Eindhoven University of Technology, P.O. Box 513, Eindhoven, The Netherlands

(Received 27 September 2012; final version received 7 May 2013)

Multi domain modelling provides great opportunities for possible synergy between the building simulation domain and other scientific and technological domains. Although domains may have quite different models, they often use common mathematical representations, based on differential algebraic equations (DAEs) and/or ordinary differential equations (ODEs). This paper reviews the use of S-Functions in SimuLink for DAEs and ODEs modelling regarding building simulation and its potential for multi domain applications. It is concluded that ODEs are directly implementable using S-Functions in SimuLink. DAEs are indirectly implementable by a manual process of integrating Dymola/Modelica models. Examples from the literature confirm the great opportunities for the combined building thermal, geothermal, electrical and grid performance simulation.

Keywords: multi domain; simulation; SimuLink

1. Introduction

Cross-industry multi-domain modelling language (CMML) represents a modelling language that is applied in several domains of Science and Technology and is commonly used by a vast number of users and developers across the industry. This provides the greatest opportunity for possible synergy between building performance in relation to other research areas such as building physics, environmental engineering, mechanical engineering, etc. Two important modelling languages that are in line with the definition of CMML are MatLab/SimuLink (2012) and Modelica (2012). SimuLink is considered as a block-oriented simulation tool. If a model is quite complex, this may lead to disorderly looking model representations in SimuLink. The main problem is that block diagram models in SimuLink are not very suitable for representing equations, such as differential algebraic equations (DAEs) and ordinary differential equations (ODEs). However, if the mathematical model is based on ODEs, so-called S-Functions can be used for computer modelling and simulation. With the use of S-Functions, the part of the code that implements the ODEs is identical to the accompanying mathematical equations. This also makes it quite easy to derive the mathematical equations from a given S-Function. Taking into account that ODEs are commonly used for modelling building systems, they provide a systematic approach for modelling and simulating building systems using SimuLink. The reader should note that the main reason for using SimuLink and

not just stand alone MatLab is the use of S-Functions which are only available in SimuLink. The main objectives of this paper are twofold, first, to show the use of S-Functions in SimuLink for DAE and ODE modelling regarding building simulation and, second, its potential for multi domain applications. The methodology consisted of three steps. For the first objective there were two steps. Step 1 was a literature study on S-functions related to SimuLink applications. Modelica allows acausal models, which in turn enable modelling using schematic editors in addition to block-diagrams in which input and output signals are connected. As SimuLink is widely used in academic and research institutions as well as industrial enterprises, Step 2 provides a study on DAE and ODE modelling using S-Functions. Because all models are implemented in SimuLink, it is expected that the coupling of multi domain SimuLink models is relative easy. The second objective is presented in Step 3 of the methodology, where several case studies from the literature will be discussed with the focus on promising model combinations. The outline of the paper is as follows: Section 2 starts with a discussion of modelling tools with the aim to position the presented simulation environment. Section 3 presents the use of S-Functions in SimuLink including an example of how an object can be developed and embedded in the building energy simulation (BES) model. Section 4 provides the CMML application potential. Section 5 shows the conclusions.

*Email: a.w.m.v.schijndel@tue.nl

2. Positioning of DAEs/ODEs based modelling

In this Section, the context of DAEs/ODEs based modelling within the domain of building performance simulation is presented. The main topics are twofold: positioning the presented approach with (1) a related modelling environment such as TRNSYS (2013), and (2) a recently developed co-simulation software environment such as the building controls virtual test bed (BCVTB), Wetter (2011a). The reader should note that the mentioned software tools, TRNSYS and BCVTB, are exemplary. An exhaustive positioning of these software tools themselves within building performance simulation can be found in Hensen and Lamberts (2011).

First, Wetter and Haugstetter (2006) published a paper on a comparison between an equation-based modelling language, Dymola Modelica (i.e. DAEs modelling language), and a procedural modelling language, TRNSYS, for BES. Among other things, Wetter and Haugstetter (2006) present a comparative model implemented and validated in both tools. Furthermore, the simulation environments were compared regarding the model development time and numerical performance. The authors found that it was easier to construct large models in Modelica than it was in TRNSYS due to the hierarchical way that Modelica facilitates debugging and the reuse of submodels. The main strength of TRNSYS is its large model libraries that have passed significant validation tests. Regarding computation time, using a single processor, Dymola computed the benchmark four times slower than TRNSYS. However, in Hensen and Lamberts (2011), Wetter states that it is reasonable to expect that models formulated in equation-based, high level programming languages, as opposed to existing simulation programmes, are in a better position to take advantage of a shift towards parallel hardware.

Second, Wetter (2011b) presents a paper on co-simulation of building energy and control systems utilizing the BCVTB. The BCVTB allows run time coupling of different simulation programmes for data exchange, including among other things MatLab, SimuLink, Modelica and Dymola. This means that DAEs/ODEs based models can be integrated into this state-of-the-art (co-)simulation environment. Another effort to facilitate co-simulation, including DAEs/ODEs based models, is the functional mockup interface (FMI) (Blochwitz et al. 2011). This interface has quite some similarities with S-Functions in SimuLink and is discussed in the next section.

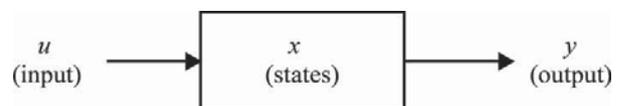
3. Introduction to S-functions in SimuLink

An independent standard for the exchange of dynamic models and for co-simulation has been researched for a long time. This is also the goal of tool independent modelling languages such as Modelica (2012). Amongst recent developments, the FMI (Blochwitz et al. 2011) appears to be an interface that is quite similar to S-Functions. Therefore,

we begin with an introduction to S-Functions and show the methodology for implementing ODEs and DAEs. After this we compare this approach with the FMI approach and in the last part of this section we provide applications for building simulation.

3.1. Description of S-functions in SimuLink

The main idea is to use S-Functions (Mathworks 1998). In order to have a good understanding of how S-Functions work, a short summary is provided here. An S-function (system-function) is a computer language description of a dynamic system available in SimuLink. The form of an S-function is very general and can accommodate continuous, discrete and hybrid systems. As a matter of fact, nearly all SimuLink models can be described as S-functions. Each block within a SimuLink model has the following general characteristics: a vector of inputs, u , a vector of outputs, y , and a vector of states, x , as shown by the illustration in Figure 1. The state vector may consist of continuous states, discrete states or a combination of both. The mathematical relationships between the inputs, outputs and the states are expressed by the equations presented in the above figure. SimuLink partitions the state vector x into two parts: the continuous states x_c and the discrete states x_d . The continuous states occupy the first part of the state vector, and the discrete states occupy the second part. For blocks with no states, x is an empty vector. SimuLink makes repeated calls during specific stages of the simulation to each block in the model, directing it to perform tasks such as computing its outputs, updating its discrete states or computing its derivatives. Additional calls are made at the beginning and end of a simulation to perform initialization and termination tasks. Figure 2 illustrates how SimuLink performs a simulation. First, SimuLink initializes the model; this includes initializing each block, including S-functions. Then, SimuLink enters the simulation loop, where each pass through the loop is referred to as a simulation step. During each simulation step, SimuLink executes



$$y = f_0(t, x, u) \quad (\text{output})$$

$$\dot{x}_c = f_d(t, x, u) \quad (\text{derivative})$$

$$x_{d_{k+1}} = f_u(t, x, u) \quad (\text{update})$$

$$\text{where } x = x_c + x_d$$

Figure 1. Illustration of the general characteristics of an S-Function.

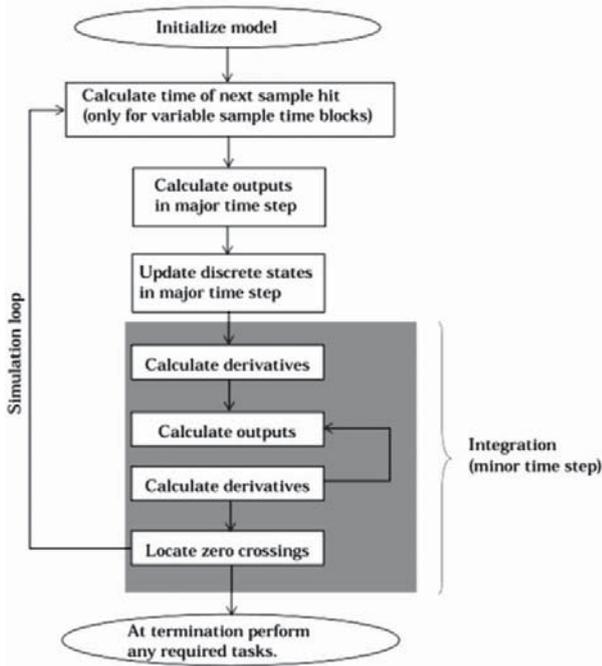


Figure 2. How SimuLink performs simulation.

the S-function block. This continues until the simulation is complete (Figure 2). SimuLink makes repeated calls to S-functions in a model. During these calls, SimuLink calls S-function routines, which perform tasks required at each stage. These tasks include: Initialization, calculation of next sample hit, calculation of outputs in the major time step, update discrete states in the major time step, and integration.

3.2. ODE and DAE modelling using S-Functions

In this Section, we show methodologies for implementing both ODE as well as DAE models in SimuLink using S-Functions.

3.2.1. Implementing ODEs using S-Functions

ODE-based models have been integrated into SimuLink using the continuous states part of an S-Function. A heat pump is used as an example. The first step is to develop a mathematical model of the heat pump, based on first principles, in the form of ODEs

$$\text{COP} = k \cdot \frac{0.5 \cdot T_{\text{cin}} + 0.5 \cdot T_{\text{cout}} + 273.15}{(0.5 \cdot T_{\text{cin}} + 0.5 \cdot T_{\text{cout}}) - (0.5 \cdot T_{\text{vin}} + 0.5 \cdot T_{\text{vout}})},$$

$$C_c \frac{dT_{\text{cout}}}{dt} = F_{\text{cin}} \cdot c_w \cdot (T_{\text{cin}} - T_{\text{cout}}) + \text{COP} \cdot E_{\text{hp}},$$

$$C_v \frac{dT_{\text{vout}}}{dt} = F_{\text{vin}} \cdot c_w \cdot (T_{\text{vin}} - T_{\text{vout}}) - (\text{COP} - 1) \cdot E_{\text{hp}},$$
(1)

Table 1. The input–output definition of the heat pump model.

Name	Input (u)/ output (y)	Description
T_{vin}	$u(1)$	Incoming water temperature at the evaporator ($^{\circ}\text{C}$)
F_{vin}	$u(2)$	Incoming mass flow at the evaporator (kg/s)
T_{cin}	$u(3)$	Incoming water temperature at the condenser ($^{\circ}\text{C}$)
F_{cin}	$u(4)$	Incoming mass flow at the condenser (kg/s)
E_{hp}	$u(5)$	Power of electrical supply (W)
k	$u(6)$	efficiency (–)
T_{vout}	$y(1) = x(1)$	Outgoing water temperature at the evaporator ($^{\circ}\text{C}$)
T_{cout}	$y(2) = x(2)$	Outgoing water temperature at the condenser ($^{\circ}\text{C}$)
COP	$y(3)$	COP (–)

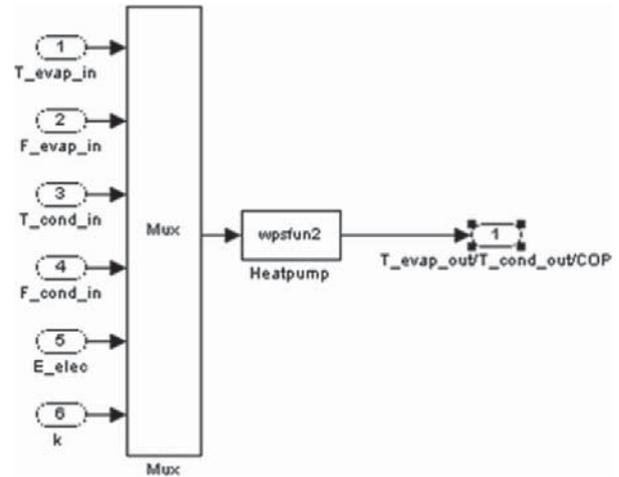


Figure 3. The input–output structure in SimuLink.

where T is the temperature ($^{\circ}\text{C}$), COP is the coefficient of performance (–), k is the heat pump efficiency (–), c_w is the specific heat capacity of water (J/kgK), C is the heat capacity of the water and pipes in the heat pump (J/K), t is the time (s), F is the mass flow (kg/s) and E_{hp} is the heat pump electric power supply (W). Subscript c means water at the condenser, v means water at the evaporator, in means incoming and out means outgoing. The second step is to prepare the input–output definition of the model. This is presented in Table 1. The third step is to build the input–output structure connected with the S-function block. Figure 3 shows the corresponding SimuLink model. The fourth step is to write code into an S-Function. The technical details of this step can be found in the appendix. The importance of these details is that the relation between the mathematical model and the implemented model is clear and is useful for developers who want to create their own ODE-based models. It should be mentioned that the time

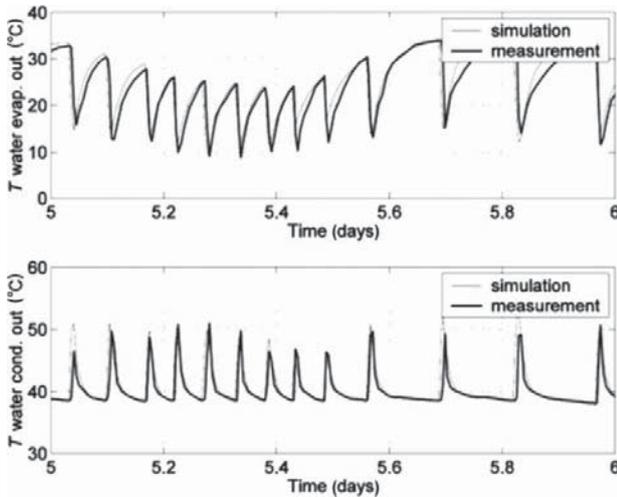


Figure 4. The measured and simulated water temperature of the evaporator (top) and condenser (bottom).

scale problem, i.e. the lowest time constant present in the dynamic system is five orders of magnitude lower than the highest one (for example, the refrigerant space), seems less relevant in the case of ODE-based models because there are special designed solvers for this case. So-called stiff solvers can handle such a problem accurately and efficiently (Ashino, Nagase, and Vaillancourt 2000). On the other hand, simulating controllers, for example (rapid), on/off cycling can generate small time steps during simulation. Although accurate results are obtained in this case, it could still lead to relatively long simulation times. A test site at the Eindhoven University was used for validation (i.e. simulated data compared with experimental data). This test site contained a configuration with the following subsystems: a heat pump, a heat exchanger on the roof (energy roof) and a thermal energy storage system. All subsystems were modelled using the modelling approach of this section. The experimental results obtained from the test site were used to validate the models of all three systems listed above. From this work, published in van Schijndel (2007), the validation results of the heat pump are presented in Figure 4. In this figure, the measured and simulated water temperatures of the outgoing flows at the condenser and evaporator are compared. The other components: energy roof and thermal energy storage were validated with similar results. It is concluded that the presented ODE-based modelling approach provides accurate results. In general, it is expected that systems modelled as a system of ODEs can be relatively easily and accurately (with a mean error less than 3%) simulated using the presented modelling approach.

3.2.2. Implementing DAEs using S-functions

DAEs are a general form of (systems of) differential equations for vector-values: $F(\dot{x}(t), x(t), t) = 0$, where ODEs $\dot{x}(t) = g(x(t), t)$. Due to the more general nature of DAEs,

they may be more useful for equation-based languages, having the properties that a modeller can express the sequence of instructions in an arbitrary way. A key feature of these languages is that they allow acausal models, which in turn enable modelling using schematic editors. It also allows manipulation of the equations for various purposes and the equations can be stated without considering computational order. According to Zupancic and Sodja (2008) this is called acausal modelling and causality is artificially made because physical laws must be transformed into convenient computational descriptions. Furthermore, Zupancic and Sodja (2008) suggest using Dymola (2013) with Modelica (2012) for acausal modelling to write balance and other equations in their natural form as a system of DAEs and in order to achieve efficient simulation code, computer algebra is utilized, similar to the implementation of ODEs. The latter was confirmed by the work of Jeganathan, Pitchaikani, and Dharumaseelan (2011) through the development of a manual process of integrating Modelica plant models into SimuLink by use of S-Functions. In Figure 5 (excerpt from Jeganathan, Pitchaikani, and Dharumaseelan 2011), the flowchart of this process is shown. We may conclude that although it is not possible to solve DAEs directly with S-Functions, it is possible to implement DAE-based models of Dymola/Modelica in S-Functions. This is also shown more recently by Zupancic and Sodja (2012).

3.2.3. Similarities with the tool independent exchange interface FMI

As mentioned before, the FMI (Blochwitz et al. 2011) appears to be an interface that is quite similar to S-Functions. The accompanying interface facilitates so-called ‘hybrid ODE’ meaning ODEs in state space (continuous system) form with events handled (discrete system). This is very similar to the general use of S-Functions that also facilitates continuous systems (see Section 2.1) and discrete systems (Mathworks 1998). An illustrative example of the hybrid use of S-Functions can be found in Section 2.3, where a whole building hourly based (discrete) simulation tool HAMBBase (de Wit 2006) has been implemented into SimuLink using hybrid (both continuous as well as discrete) S-Functions, called HAMBBase_S.

We proceed with the *mathematical description* of FMI and S-Functions. We use the terminology of Blochwitz et al. (2011) because it is more general than the description of S-Functions. This type of system is described as a piece-wise continuous system. The state of a hybrid ODE is represented by both a time-continuous state called $\mathbf{x}(t)$ in FMI and mdlDerivatives in S-Functions as well as a time-discrete state $\mathbf{m}(t)$ in FMI and mdlUpdates in S-Functions. At every event instant, t_i , variables might be discontinuous and therefore have two values at this time instant, the ‘left’ and the ‘right’ limit. The states $\mathbf{x}(t_i)$ and $\mathbf{m}(t_i)$ are always defined to be the right limit at t_i , whereas $\mathbf{x}^-(t_i)$, $\mathbf{m}^-(t_i)$ are defined to be the ‘left’ limit at t_i , e.g.: $\mathbf{m}^-(t_i) = \mathbf{m}(t_i - 1)$.

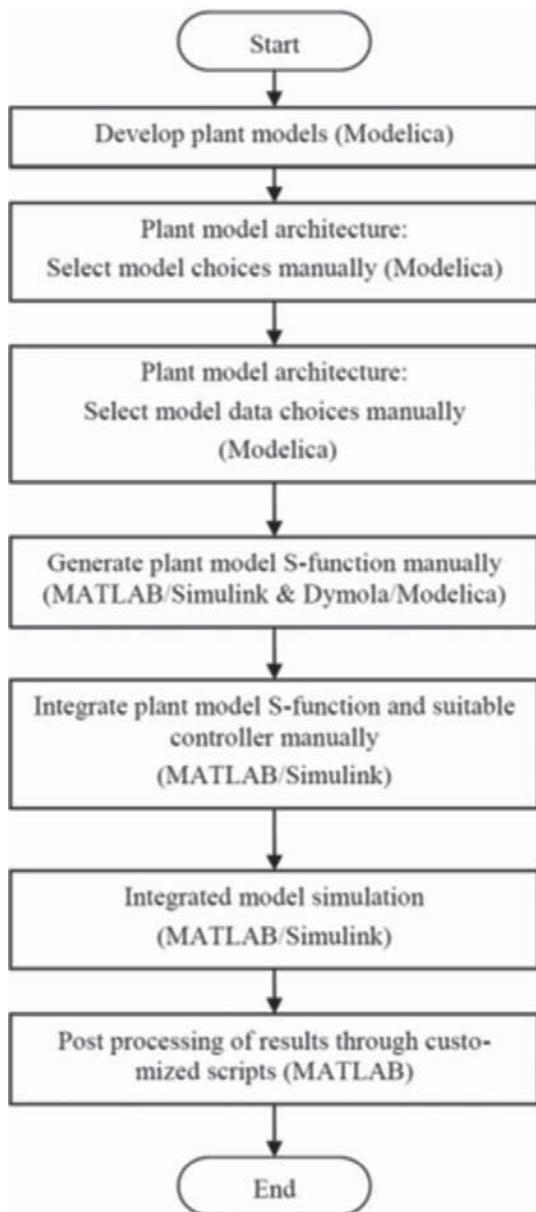


Figure 5. Flowchart of the manual process of integrating Modelica plant models into Simulink by the use of S-Functions. Source: excerpt from Jeganathan, Pitchaikani, and Dharumaseelan (2011).

The two variable types are visualized in Figure 6 (excerpt from Blochwitz et al.). For this part S-Functions are similar with one exception: $x(t)$ is always continuous, i.e. $x^-(t_i) = x(t_i)$. Using FMI an event instant t_i is defined by one of the following three conditions:

- (1) At a predefined time instant by the time-discrete state.
- (2) At every completed step of the integrator.
- (3) At a time instant where an event indicator switches on/off.

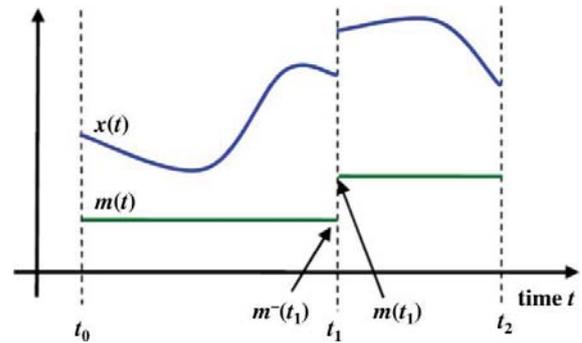


Figure 6. Piece-wise-continuous states of an FMI: time-continuous (x) and time-discrete (m). Source: excerpt from Blochwitz et al. (2011).

These conditions are also implemented in S-Functions except condition (3) the event indicator. The latter is not present in S-Functions but in complete Toolboxes Stateflow (MatLab 2012). SimEvents (MatLab 2012) are dedicated to the functionality of events based modelling.

Computation efficiency of FMI and S-Functions. Both FMI as well as S-Functions are designed so that the interfaces require only the computation of variables that are needed. For example, it is important that in the call to compute a discrete step, the state derivatives are not newly computed again, if they already exist at the present time. Both interfaces (FMI and S-Functions) decide internally and automatically which part of the model code is evaluated in the present context.

3.3. Building energy and indoor climate modelling using S-Functions

A whole building model has been integrated into Simulink using both the continuous states part as well as the discrete states part of an S-Function. The whole building model originates from the thermal indoor climate model ELAN which was already published in 1988 (de Wit and Driessen 1988). Separately a model for simulating the indoor air humidity (AHUM) was developed. In 1992, the two models were combined (WAVO) and programmed in the MATLAB environment (van Schijndel and de Wit 1999). Since that time, the model has constantly been improved using the newest techniques provided by recent MATLAB versions. Currently, the hourly based model named HAMBBase is capable of simulating the indoor temperature, the indoor AHUM and energy use for heating and cooling of a multi-zone building. The physics of this model is extensively described by de Wit (2006). The HAMBBase model uses an integrated sphere approach. It reduces the radiant temperatures to only one node which gives if the advantage of being able to easily model complicated geometries. The HAMBBase model was implemented in an S-Function by splitting the energy and vapour flows into two parts. First, a discrete part was developed for modelling 'slow responses', i.e. the transmittance

through walls. Second, a continuous part was developed for the ‘fast responses’, i.e. admittance, ventilation, heat gains, etc. Detailed information can be found in de Wit (2006). This new model, named HAMBBase_S (abbreviation for HAMBBase model in SimuLink) to avoid confusion, is able to simulate complicated heating, ventilation, and air conditioning (HVAC) installations and controls simultaneously with the building. The main advantages are:

- (a) The dynamics of the building systems where small time scales play an important role (for example, on/off cycling) are accurately simulated.
- (b) The model becomes time efficient because the computation of the ‘slow response’ part of the model only takes place at an hourly intervals and not on every (much smaller) time step of the ‘fast response’ part of the model.
- (c) The moisture (vapour) transport model of HAMBBase_S is also included and with this feature, the (de-) humidification of HVAC systems can also be simulated.

Therefore, the main difference between HAMBBase and HAMBBase_S is the time step. HAMBBase has fixed step sizes of one hour. HAMBBase_S has variable step sizes (ranging from smaller than a second to 15 min) depending on the model parameters. The codes in the S-Functions used for ODEs are quite easy to comprehend meaning that the relation between the implemented code and the mathematical model is clear and the code itself is concise. However, the code of the whole building model in the S-Function is much more complex. In order to gain confidence in the modelling, i.e. HAMBBase_S, two benchmark studies are presented now for the evaluation of the modelling approach using S-Functions. We proceed with *verification* studies, i.e. comparison with simulation results of other (analytical) models.

Verification of the thermal part: The test described here was presented in IEA Annex 41 (2008). For this test, four cases were selected from the ASHRAE Bestest. The Bestest (ASHRAE 2001) is a set of well-documented test cases for software-to-software comparisons and programme diagnostics. Good results make it likely that there are no severe internal (programming) errors and that the assumptions made are justified. The cases selected were: Case 600FF: lightweight structure, base case, free floating temperature; Case 900FF: heavyweight structure, base case, free floating temperature; Case 600: lightweight structure, base case, heating and cooling system and Case 900: heavyweight structure, base case, heating and cooling system. The results of HAMBBase_S fall mostly within the range given in the Bestest document (ASHRAE 2001). The peak loads and extreme temperatures appear at the same appropriate time (hour, day and month). This might be surprising as:

- (a) The distribution of solar radiation in the room is by the integrating sphere method and not by the Bestest prescribed distribution.

- (b) The free-floating cases are sensitive for the interior surface coefficient for convection which is constant in HAMBBase_S and also independent of the heat flow direction.
- (c) The exterior surface coefficient is the default HAMBBase_S value and not the wind dependent coefficient of the Bestest.
- (d) The dependence of the glazing absorptance on the incident angle is not modelled (only the reflectivity).

In conclusion, 11 results (85%) fit within the required bandwidth, 2 results (15%) lie outside the range.

Verification of the hygric part: In IEA Annex41 (2008) two cases were proposed for analytical solution and solution of the building model: (i) Case 0A: Isothermal exposure with vapour tight construction surfaces and (ii) Case 0B: Isothermal exposure with vapour open construction surfaces. Compared with the Bestest cases, the following changes are made for the new cases to include (simplified) vapour transport:

- (1) Constructions are made of monolithic aerated concrete with constant/linear properties.
- (2) Tight membranes on the outside, and in Case 0A also on the inside, prevent loss of vapour from the building by transport all the way through the walls.
- (3) The exposure is completely isothermal, i.e. the same temperature is maintained inside and outside the building.
- (4) The building has no windows;
- (5) Initial conditions consist of a temperature of 20°C and relative humidity (RH) of 30%. These are also the initial conditions of materials in the constructions.
- (6) Internal moisture gain = 500 g/h from 9:00 to 17:00 every day with no moisture gains outside these hours.
- (7) No heat gains at any time.
- (8) Constant ventilation of 0.5 ACH.

The exact results of Bednar and Hagentoft (2005) are compared with the numerical solutions of both HAMBBase (fixed hourly time steps) as well as HAMBBase_S (variable sub hourly time steps). In Figure 7, the results for Case 0A are presented. The difference with the exact solution of Bednar and Hagentoft (2005) is not visible. This means that the solution of the second-order equation (the mass of the room air is the only hygric capacitance) is close to the exact one and errors of less than 0.05% are made in the HAMBBase_S model. In Figure 8, the result for Case 0B with the monolithic aerated concrete construction is presented. In HAMBBase_S a second-order network is derived from the diffusion equations of the walls, so there is an approximation. One can see that both models HAMBBase and HAMBBase_S deviate from the exact solution by 0.5%.

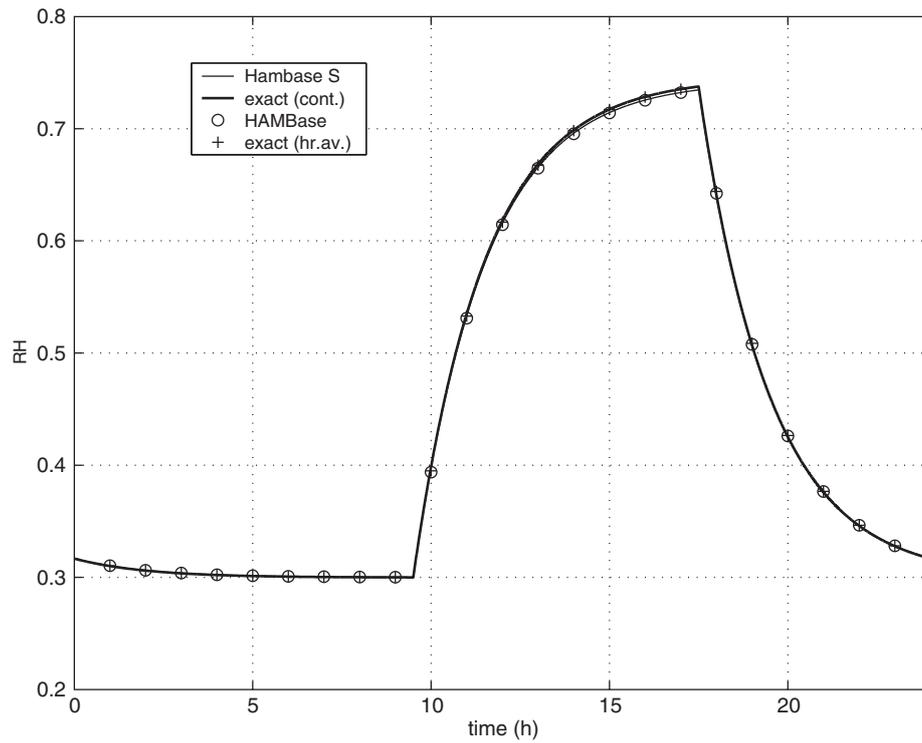


Figure 7. Simulated RH of the indoor climate with only moisture buffering by air.

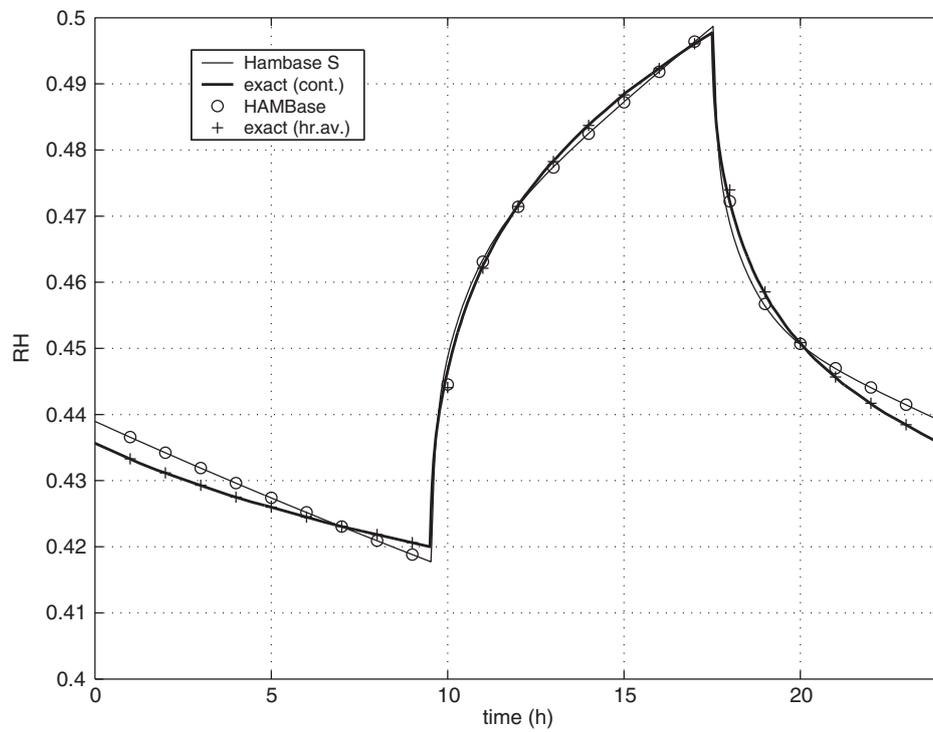


Figure 8. Simulated RH of the indoor climate with moisture buffering by air and the construction.

Table 2. Overview of papers including validation results of HAMBBase.

Study reference	Project/building	Zones (#)	Volume(s) (m ³)	Internal gains	Control
Schellen (2002)	Church (large)	2	18,850	Visitors	Floor heating
van Schijndel (2007)	Dwelling	3	8	None	Heating
van Schijndel et al. (2008)	Museum	1	10	Visitors	Heating/cooling
van Schijndel, Schellen, and de Wit (2009)	Church	2	3000	Visitors	Air heating
van Schijndel, Schellen, and de Wit (2009)	Annex 41 rooms	3	10	Equipment	Electric heating
Lony, van Schijndel, and Schellen (2010)	Display case	1	< 1	None	Full/free floating
van Schijndel, Schellen, and de Wit (2010)	Depot	8	1000	None	Full air-conditioning
Huijbregts et al. (2012)	Castle (seventeenth century)	7	250	300–500 W	Conservational heating
	Castle (thirteenth century)	3	400	Visitors	Free floating

Validation of the combined thermal-hygric model (experimental proof): The HAMBBase and HAMBBase_S models have successfully been used since 2003 for amongst other things monumental buildings and museums. These types of buildings are very interesting for validation purposes because not only energy usage and thermal comfort are important but also the indoor climate and especially the relative humidity are crucial for conservation of artefacts. Table 2 provides an overview of papers, including successful (experimental) validation results.

3.4. A worked example to couple ODEs based models with HAMBBase_S in SimuLink

Section 3.2.1 already showed how an object (heat pump) starting from an equation set (1) can be converted into a SimuLink block (Figure 3). This worked example shows how the heat pump model block can be coupled with HAMBBase_S and completed with an on/off controller in SimuLink. Figure 9 (top) presents the SimuLink subsystem blocks of the heat pump and HAMBBase_S. The heat pump subsystem is created in SimuLink by selecting all blocks of Figure 3 and then creating a subsystem. The HAMBBase_S subsystem is automatically created from the HAMBBase input mfile (see Section 3.3). In this example, a single zone was modelled, so the input of the HAMBBase_S subsystem consists of one heat load and one moisture load to the zone. The output consists of the zone temperature and relative humidity. The coupling of the models of the heat pump and HAMBBase_S is straight forward. There are three main couplings as shown in Figure 9. First, the heat load input of HAMBBase_S is coupled with the output (produced heat) of the heat pump. Second, the electrical power input of the heat pump is coupled with the output of the on/off controller. The latter is a relay providing, respectively, 1000 W of heating power if the input is below 19°C and 0 W if the input is above 21°C. Third, the input of the relay is coupled

with the output of the HAMBBase_S block, i.e. the temperature inside the zone. All remaining inputs are connected with constant blocks providing the necessary parameters. This completes the building of the model in SimuLink. The remainder is the initialization of the simulation. The results are shown in Figure 4.

3.4.1. Applications

Table 2 also provides applications of the example outlined above.

3.5. Alternative method for implementing DAEs models in SimuLink

As shown above, S-Functions in SimuLink are suitable for ODEs and not for DAEs. However, there exists an alternative method for implementing DAEs in SimuLink using Simscape (2011), a toolbox for physical modelling that was developed by Mathworks. The language is based on MatLab syntax, but it is possible to define acausal models. Sjostedt (2009) compares similar physical models in Dymola–Modelica and Simscape in a mechatronic context. The main conclusions of Sjostedt (2009) regarding the comparison between Simscape and Modelica are that the modelling languages are quite similar and the equations have well-defined semantics although the syntax varies between languages. Converting component models from Modelica to Simscape is possible and is a second alternative for importing Modelica models into SimuLink (see Section 3.2.2). Vice versa is also possible, however, today not too many Simscape components are available for conversion to Modelica.

4. CMML applications using SimuLink

This section presents the CMML applications using SimuLink for the following categories: building simulation, power supply simulation and multi-domain simulation. The aim is to show representative tools to support the

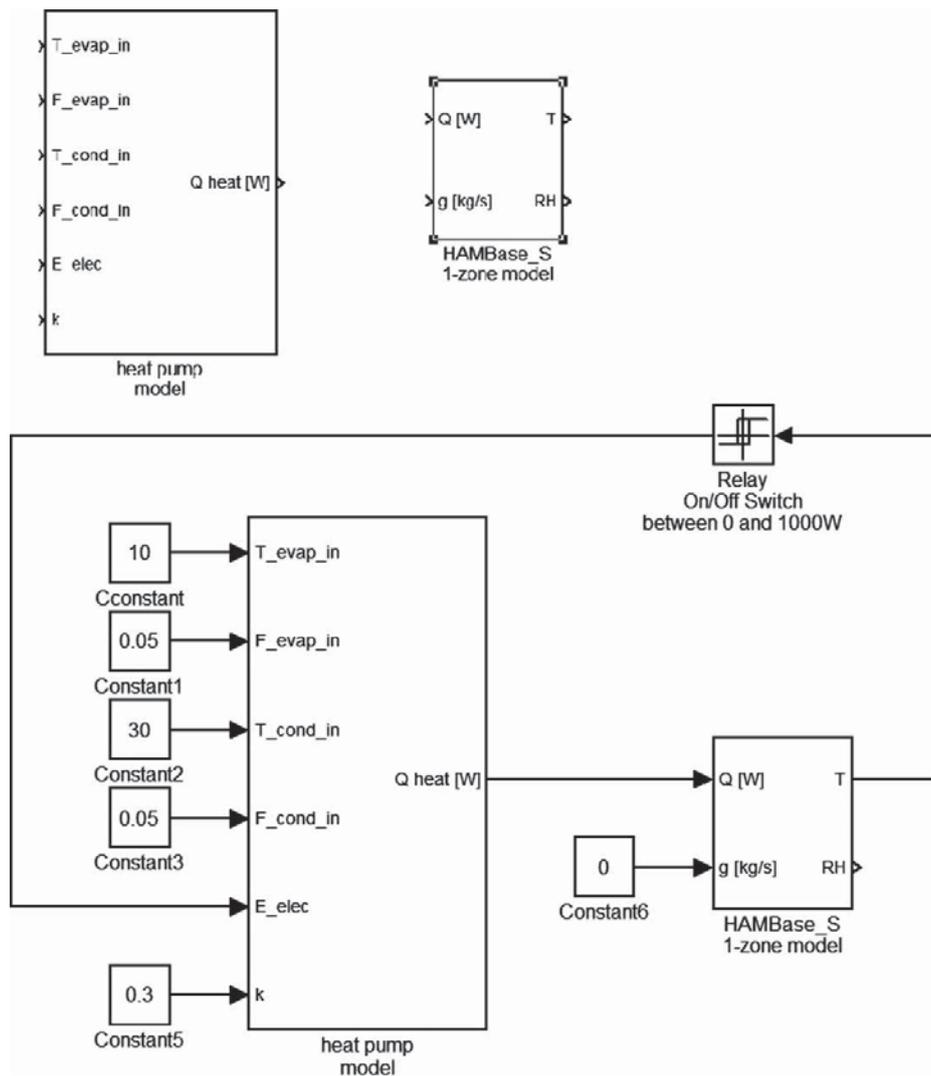


Figure 9. Worked example on coupling the heat pump model with HAMBBase_S in SimuLink. Top: The individual models of the heat pump and HAMBBase_S. Bottom: The models coupled and completed with constants and an on/off controller

methodology of this paper and the following is not intended as a complete list.

4.1. Building simulation tools with the focus on heat air and moisture transport

Riederer (2005) provided a state-of-the-art description for MatLab/SimuLink tools for building and HVAC simulation. More recently, Orosa (2011) presented a new modelling methodology to control HVAC systems, including the use of SimuLink. This work argues in favour of open and freely available heat air and moisture (ham) tools in SimuLink. The requirements on simulation tools in SimuLink, presented in Orosa (2011), will also be adopted in this paper. Two toolboxes that meet these requirements are the International Building Physics Toolbox (IBPT; Sasic

Kalagasidis et al. 2007) and the Heat Air and Moisture Laboratory (HAMLab; van Schijndel 2007).

HAMLab (2012) evolved from the HAMBbase modelling presented in Section 2.3. A recent overview can be found in van Schijndel (2009). The paper concludes that a large range of integrated HAM problems can be solved using the SimuLink/COMSOL environment (i.e. co-simulation of SimuLink and COMSOL 2012). Furthermore, it seems promising for accurately solving modelling problems that are caused by the presence of different time scales and/or lumped/distributed parameters. The main new development is to integrate models regarding the different spatial scales of \sim mm (material physics); \sim m (building physics); \sim km (urban physics) and \sim Mm (global climate physics).

IBPT (2012) is primarily developed for research purposes. A graphical programming environment and the

ability to build complex models within a short time also makes it useful for education in the area of building physics. The interaction between various parts of the model is presented for a user through a graphical environment in SimuLink, and, furthermore, it is possible to develop new models and obtain results fairly quickly. Currently, the IBPT mainly contains models of heat, air and moisture transport in building constructions; indoor air pressure and walls with dynamic insulation; volatile organic compounds (VOCs) in indoor air; contamination of concrete with VOCs; hybrid 2D/3D floor heating and ventilated attics.

4.2. Power supply simulation examples

In this section two papers, including power supply tools that are relevant for CMML applications and were developed in SimuLink, are briefly discussed. Figures of the characteristic systems are excerpts from the original papers.

Panahandeh et al. (2011) present the simulation of photovoltaic-wind-hybrid systems combined with hydrogen storage for rural electrification. The hybrid energy system consists of energy storage and power conditioning equipment. Figure 10 shows the basic designs of this system (excerpt from Panahandeh et al. 2011). Figure 10(a) provides a DC-coupled system where all generators are coupled on a DC bus. The power delivered by all energy sources is controlled by a central unit connected to the distribution grid via a single point. Figure 10(b) presents an AC-coupled system where distributed systems are coupled directly via the distribution grid. The following models have been integrated into SimuLink as S-Functions:

- (1) photovoltaic-generator (PVG) and wind energy converter;
- (2) battery;

- (3) hydrogen path based on fuel cell, electrolyser, compressor and gas bottles and
- (4) inverters.

Details of each model can be found in Panahandeh et al. (2011). The simulation environment developed under SimuLink was successfully used to:

- (a) simulate the realistic operation of the hybrid system;
- (b) determine the influence of inverter parameters on the operation;
- (c) adjust the optimal parameters of inverters and
- (d) integrate a hydrogen storage path in the hybrid system operation.

Furthermore, Panahandeh et al. (2011) conclude that making use of the existing model library dynamic simulations can be an efficient means for very detailed analysis of the behaviour and performance of the components and the hybrid system as a whole.

Skretas and Papadopoulos (2009) present the simulation of a photovoltaic-wind-hybrid system with voltage regulation features in compliance with electric grid requirements. Figure 11 shows schematically the grid connected hybrid wind turbine and photovoltaic power system (GCHW-PPS) configuration (excerpt from Skretas and Papadopoulos 2009). It consists of

- (1) PVG, two Buck-Boost DC-DC converters (BBC1, BBC2) and a control unit including a digital signal processor (DSP1);
- (2) wind turbines (WTs), two Buck-Boost DC-DC converters (BBC3, BBC4) and a control unit DSP2;
- (3) power decoupling capacitor (C_{PD}) and coupling unit and

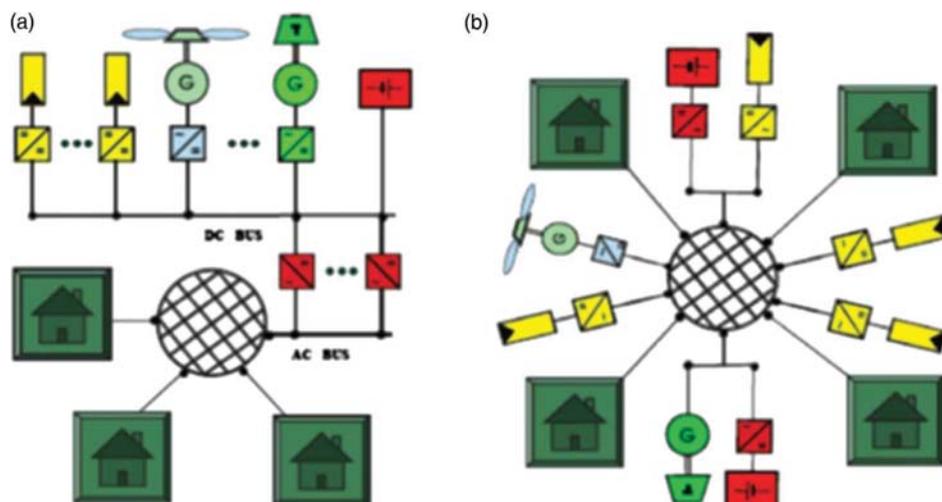


Figure 10. (a) DC-coupled (right) hybrid system and (b) AC-coupled hybrid system. Source: excerpt from Panahandeh et al. (2011).

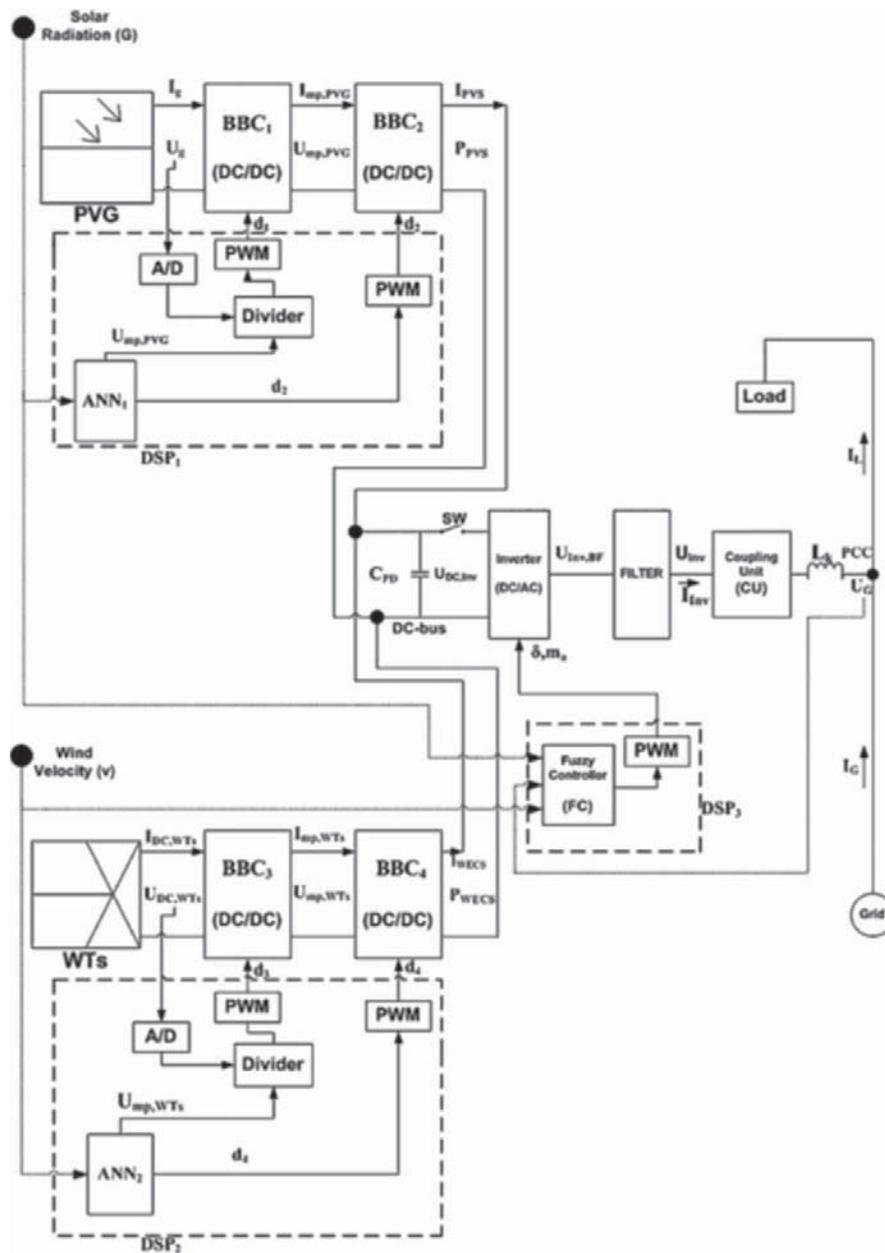


Figure 11. Simplified general representation of the proposed GCHWPPS configuration.

Source: Reprinted from *Electric Power Systems Research*, Vol. 79, Issue 9, S.B. Skretas and D.P. Papadopoulos, *Efficient design and simulation of an expandable hybrid (wind-photovoltaic) power system with MPPT and inverter input voltage regulation features in compliance with electric grid requirements*, pp. 1271–1285, Copyright 2009, with permission from Elsevier.

- (4) a fuzzy logic control unit DSP3 for selecting optimal parameters.

SimuLink/MATLAB was used to model and integrate all the above presented components for the simulation and investigation of the examined GCHWPPS. Skretas and Papadopoulos (2009) conclude that the results of the conducted simulations of the examined GCHWPPS (with rated power 4255 W) showed that this system configuration along with the proposed intelligent control techniques

achieve rapidly, accurately and simultaneously: maximum power of the WTs and PVG configuration, DC voltage regulation/stabilization at the input of the inverter, and high quality of the power being injected to the grid.

4.3. Multi domain applications

In this section, two multi domain applications are presented consisting of a combined building performance and another domain using SimuLink as the modelling tool.

The first multi domain application consists of a combined building thermal and electrical performance simulation. The latter includes electricity production and consumption of buildings. Salvador and Grieu (2012) present a methodology for the design of energy production and storage systems in buildings in order to minimize the energy impact on the electricity grid. The modelling includes:

- (1) single storey houses;
- (2) solar thermal panels, photovoltaic panels and a vertical axis WT integrated to the buildings;
- (3) a thermal storage tank used for domestic hot water and
- (4) occupancy scenarios and a ventilation controller.

Salvador and Grieu (2012) conclude that energy self-consumption allows for a more efficient and rational use of energy, as it would imply significant savings in electricity distribution and transport costs and a reduction in the need to invest in new networks. The reader should note that the models presented in this work are just partly based on SimuLink. However, regarding the work of Riederer, Keilholz, and Ducreux (2009), i.e. the coupling of TRNSYS with SimuLink, it should be possible to completely integrate the above-mentioned models into SimuLink.

The second multi domain application consists of a combined building and geothermal performance simulation. Partenay et al. (2011) present a simulation of the influence of the borehole short time response on ground source heat pump system efficiency. The work includes:

- (1) development of a ground heat exchanger model including the dynamic phenomena inside the borehole;
- (2) experimental study of the ground heat exchangers in open loop configuration;
- (3) coupling building simulation with experimental ground heat exchanger (closed loop configuration) and
- (4) parametric study using annual simulations.

In this case, ground heat exchanger models of TRNSYS were exported to the SimuLink environment by the method of Riederer, Keilholz, and Ducreux (2009). The integrated model in SimuLink is able to simulate the operation of a real borehole in a closed loop with a perfectly controlled building. Its main purpose is to compare the relative difference between steady-state and transient borehole models. Partenay et al. (2011) conclude that for one climate the relative difference between steady-state and transient borehole models ranges from 0% to 8% and 0% to 18% for heating and cooling cases, respectively.

5. Conclusions

This paper presents the use of S-Functions in SimuLink for DAEs and ODEs modelling regarding building simulation and its potential for multi domain applications. It is concluded that ODEs are directly implementable using S-Functions in SimuLink. DAEs are indirectly implementable by a manual process of integrating Dymola Modelica models. Examples from the literature confirm the great opportunities for the combined building thermal, geothermal, electrical and grid performance simulations.

The author expects that one of the main future developments of equation-based modelling for building performance simulation will be the integration of partial differential equations. Particularly, BES in combination with 3D computational modelling (fluid dynamics, heat, air and moisture transport) is still a challenge for equation-based modelling.

Acknowledgements

John Bynum is greatly acknowledged for his contribution.

References

- Ashino, R., M. Nagase, and R. Vaillancourt. 2000. "Behind and Beyond the MATLAB ODE Suite." *Computers & Mathematics with Applications* 40 (4), 491–512.
- ASHRAE. 2001. *Standard Method of Test for the Evaluation of Building Energy Analysis Computer Programs*. Vol. 140. Richmond, VA: ASHRAE.
- Bednar, T., and C. E. Hagentoft. 2005. "Analytical Solution for Moisture Buffering Effect; Validation Exercises for Simulation Tools." Nordic symposium on building Physics, Reykjavik, June 13–15, 2005, 625–632.
- Blochowitz, T., M. Otter, M. Arnold, C. Bausch, C. Clauß, H. Elmqvist, A. Junghanns, et al. 2011. "The Functional Mockup Interface for Tool Independent Exchange of Simulation Models." Proceedings 8th Modelica conference, Dresden, March 20–22, 2011, 105–114.
- COMSOL Website. 2012. Accessed January, 2013. <http://www.comsol.com/>.
- Dymola Website. Accessed January, 2013. <http://www.3ds.com/products/catia/portfolio/dymola>.
- HAMLab Website. 2012. Accessed January, 2013. <http://archbps1.campus.tue.nl/bpswiki/index.php/Hamlab>.
- Hensen, J. L. M., and R. Lamberts (2011). *Building Performance Simulation for Design and Operation*. Abingdon, UK: Routledge. ISBN: 978-0-415-47414-6.
- Huijbregts, Z., R. P. Kramer, M. H. J. Martens, A. W. M. van Schijndel, and H. L. Schellen. 2012. "A Proposed Method to Assess the Damage Risk of Future Climate Change to Museum Objects in Historic Buildings." *Building and Environment* 55 (1), 43–56.
- IBPT Website. 2012. Accessed January, 2013. <http://www.ibpt.org/>
- IEA Annex 41. 2008. "Whole Building Heat, Air, Moisture Response." In *Modeling Principles and Common Exercises*, 1st ed., edited by M. Woloszyn and C. Rode, chap. 4, 153–169. Leuven: KU Leuven.
- Jeganathan, E. J., A. Pitchaikani, and E. Dharumaseelan. 2011. "Productivity Improvement Tool for Configuration of Modelica Plant Models and Integration with Simulink Controller Models." Proceedings 8th Modelica conference, Dresden, March 20–22, 2011, 865–871.

- Lony, R. J. M., A. W. M. van Schijndel, and H. L. Schellen. 2010. "Evaluation of the Climate Control Performance and Reliability of Active Display Cases." *Restoration of Buildings and Monuments* 16 (1): 15–26.
- Mathworks, Inc. 1998. "Writing S-Functions." Accessed January, 2013. www.mathworks.com
- Matlab, Website. 2012. Accessed January, 2013. <http://www.mathworks.com/>
- Modelica, Website. 2012. Accessed January, 2013. <https://modelica.org/>
- Orosa, J. A. 2011. "A New Modelling Methodology to Control HVAC Systems." *Expert Systems with Applications* 38 (1): 4505–4513.
- Panahandeh, B., J. Bard, A. Outzourhit, and D. Zejli. 2011. "Simulation of PVEWind-Hybrid Systems Combined with Hydrogen Storage for Rural Electrification." *International Journal of Hydrogen Energy* 36 (1): 4185–4197.
- Partenay, V., P. Riederer, T. Salquea, and E. Wurtzb. 2011. "The Influence of the Borehole Short-Time Response on Ground Source Heat Pump System Efficiency." *Energy and Buildings* 43 (1): 1280–1287.
- Riederer, P. 2005. "MatLab/SimuLink for Building and HVAC Simulation – State of the Art." 9th international IBPSA conference, Montreal, August 15–18, 2005, 1019–1026.
- Riederer, P., W. Keilholz, and V. Ducreux. 2009. "Coupling of TRNSYS with SimuLink – A Method to Automatically Export and Use TRNSYS Models Within SimuLink and Vice Versa." 11th international IBPSA conference, Glasgow, July 27–30, 2009, 1628–1633.
- Salvador, M., and S. Grieu. 2012. "Methodology for the Design of Energy Production and Storage Systems in Buildings: Minimization of the Energy Impact on the Electricity Grid." *Energy and Buildings* 47 (1): 659–673.
- Sasic Kalagasidis, A., P. Weitzmann, T. Rammer Nielsen, R. Ruut Peuhkuri, C.-E. Hagentoft, and C. Rode. 2007. "The International Building Physics Toolbox in Simulink." *Energy and Buildings* 39 (1): 665–674.
- Schellen, H. L. 2002. "Heating Monumental Churches." PhD diss., Eindhoven University of Technology.
- van Schijndel, A. W. M. 2007. "Integrated Heat Air and Moisture Modeling and Simulation." PhD diss., Eindhoven University of Technology.
- van Schijndel, A. W. M. 2009. "Integrated Modeling of Dynamic Heat, Air and Moisture Processes in Buildings and Systems Using SimuLink and COMSOL." *Building Simulation: An International Journal* 2 (2): 143–155.
- van Schijndel, A. W. M., H. L. Schellen, and W. J. Timmermans. 2010. "Simulation of the Climate System Performance of a Museum in Case of Failure Events." *Energy and Buildings* 42 (10): 1790–1796.
- van Schijndel, A. W. M., H. L. Schellen, J. L. Wijffelaars, and K. van Zundert. 2008. "Application of An Integrated Indoor Climate, HVAC and Showcase Model for the Indoor Climate Performance of a Museum." *Energy and Buildings* 40 (4): 647–653.
- van Schijndel, A. W. M., H. L. Schellen, and M. H. de Wit. 2009. "Improved HVAC Operation to Preserve a Church Organ." *Building and Environment* 44 (1): 156–168.
- van Schijndel, A. W. M., and M. H. de Wit. 1999. "A Building Physics Toolbox in MatLab." 7th symposium on Building Physics in the Nordic Countries, Goteborg, 81–88.
- Simscape, 2011. *Simscape 3 User's Guide*. Accessed January, 2013. www.mathworks.com
- Sjostedt, C.-J. 2009. "Modeling and Simulation of Physical Systems in a Mechatronic Context." Doctoral diss., KTH Sweden.
- Skretas, S. B., and D. P. Papadopoulos. 2009. "Efficient Design and Simulation of an Expandable Hybrid (Wind-Photovoltaic) Power System with MPPT and Inverter Input Voltage Regulation Features in Compliance With Electric Grid Requirements." *Electric Power Systems Research* 79 (1): 1271–1285.
- TRNSYS, Website. 2013. Accessed January, 2013. www.trnsys.com
- Wetter, M. 2011a. "A View on Future Building System Modeling and Simulation Book." In *Building Performance Simulation for Design and Operation*, edited by L. M. Hensen and Roberto Lamberts, chap. 17, 481–504. London, UK: Routledge. ISBN: 978-0-415-47414-6.
- Wetter, M. 2011b. "Co-Simulation of Building Energy and Control Systems with Building Controls Virtual Test Bed." *Journal of Building Performance Simulation* 4 (3): 185–203.
- Wetter, M., and C. Haugstetter. 2006. "Modelica Versus TRNSYS – A Comparison Between an Equation-Based and a Procedural Modeling Language for Building Energy Simulation." Proceedings of SimBuild 2006, Cambridge, 262–269.
- de Wit, M. H. 2006. *HAMBase, Heat, Air and Moisture Model for Building and Systems Evaluation*. Eindhoven: Eindhoven University of Technology, Faculteit Bouwkunde.
- de Wit, M. H., and H. H. Driessen. 1988. "ELAN A Computer Model for Building Energy Design." *Building and Environment* 23 (1): 285–289.
- Zupancic, B., and A. Sodja. 2008. "Thermal Conditions in Buildings-Advances of Acausal OO Modelling in Comparison with Traditional Approach." Proceedings of the 7th WSEAS international conference on System Science and Simulation in Engineering (ICOSSSE'08), Venice, 173–178.
- Zupancic, B., and A. Sodja. 2012. "Computer-Aided Physical Multi-Domain Modelling: Some Experiences from Education and Industrial Applications." *Simulation Modelling Practice and Theory* 33 (1): 45–67.

Appendix

```
function [sys,x0,str,ts] = wpsfun2
(t,x,u,flag)

%WPSFUN2 Heat pump

%u(1) = Tvin [oC], u(2)=Fvin [kg/s],
u(3)=Tcin [oC],

%u(4)=Fcin [kg/s], u(5)=Ehp [W], u(6)=k
[-]

%y(1)=Tvout [oC] (=x(1)),
y(2)=Tcout [oC] (=x(2))

%y(3)=COP

function [sys,x0,str,ts]=mdlInitializeSizes

sizes.NumContStates = 2; % Number of
Cont. states

sizes.NumDiscStates = 0; % Number of
Disc. states

sizes.NumOutputs = 3; % Number of
Outputs
```

```

sizes.NumInputs = 6; % Number of Inputs
x0 = [10; 10]; % Initial values
function sys=mdlDerivatives(t,x,u)
Tvm=(u(1)+x(1))/2;
Tcm=(u(3)+x(2))/2;
COP=u(6)*(273.15+Tcm)/(Tcm-Tvm);
Cc=200000;Cv=200000;cv=4200;
cc=4200;
xdot(1)=(1/Cv)*(u(2)*cv*(u(1)-x(1))-(COP-1)*u(5));
xdot(2)=(1/Cc)*(u(4)*cc*(u(3)-x(2))+COP*u(5));
sys = [xdot(1); xdot(2)];
function sys=mdlOutputs(t,x,u)
Tvm=(u(1)+x(1))/2;
Tcm=(u(3)+x(2))/2;
COP=u(6)*(273.15+Tcm)/(Tcm-Tvm);
sys = [x(1); x(2); COP];

```