BRCM Matlab Toolbox: Model generation for model predictive building control

Published: *Proc. American Control Conf.*, pp. 1063 - 1069, Portland, OR, USA, 4-6 June, 2014.
doi: 10.1109/ACC.2014.6858967

BRCM Matlab Toolbox: Model Generation for Model Predictive Building Control

David Sturzenegger[1], Dimitrios Gyalistras[1,2], Vito Semeraro[1], Manfred Morari[1] and Roy S. Smith[1]

***Abstract*— Model predictive control (MPC) is a promising alternative in building control with the potential to improve energy efficiency and comfort and to enable demand response capabilities. Creating an accurate building model that is simple enough to allow the resulting MPC problem to be tractable is a challenging but crucial task in the control development.**
In this paper we introduce the Building Resistance-Capacitance Modeling (BRCM) Matlab Toolbox that facilitates the physical modeling of buildings for MPC. The Toolbox provides a means for the fast generation of (bi-)linear resistance-capacitance type models from basic building geometry, construction and systems data. Moreover, it supports the generation of the corresponding potentially time-varying costs and constraints. The Toolbox is based on previously validated modeling principles. In a case study a BRCM model was automatically generated from an EnergyPlus input data file and its predictive capabilities were compared to the EnergyPlus model. The Toolbox itself, the details of the modeling and the documentation can be found at www.brcm.ethz.ch.

I. Introduction

The application of model predictive control (MPC) for the control of heating, cooling, ventilation and blind positioning in buildings has recently gained much attention within the control community, see e.g. [1], [2], [3], [4], [5], [6], [7], [8]. Compared to conventional control approaches, most studies report significant potentials for energy savings and often also comfort improvements due to the systematic integration of all actuators and their interactions as well as weather and occupancy forecasts.

In the research project OptiControl-II (2011-2013) [6], [8], we implemented MPC to control over several months the heating, cooling, air conditioning and blinds positioning of a fully occupied, well instrumented Swiss office building. The project clearly showed that the generation of a suitable model is the most challenging and time-consuming task in the development of the MPC controller. This was also the finding of other implementations of MPC on real buildings, e.g. [9]. However, for MPC to become an interesting alternative for wide-spread commercial use, the modeling effort must be small. Hence, there is great need for reliable and efficient methods for generating MPC suitable models of buildings.

Two principal ways exist for modeling buildings: identification and physical based approaches. While the former has its benefits, it is due to time and building usage constraints often highly impractical or even impossible to excite buildings sufficiently for the identification of multi-input multi-output building models. Due to this and other drawbacks, we advocate the use of physical based models in building MPC together with an online adaptation of a few parameters which are usually related to the faster dynamics of the model.

In the literature on simplified building modeling, the main approach is to model zones (spaces assumed to have uniform temperature) and building elements (walls, floors, ceilings and internal masses) linearly using lumped states that describe their temperature, and calculate resistances and heat capacities that define the heat transfer between them. In [10] we showed that: i) using such a resistance-capacitance (RC) modeling approach it is possible to separate the generation of the buildings' *thermal model* (describing the heat transfer between zones, walls and ceilings) from modeling *external heat fluxes* (e.g. solar gains, building systems, internal gains etc.); ii) the thermal model generation can be automated and parameterizable sub-models of the external heat fluxes (EHF) can be modularly added; iii) in a test case the resulting model agreed well with the original EnergyPlus[1] model it was derived from. The model used in the MPC implemented within the OptiControl-II project based on this approach and resulted in a very good control performance throughout the whole experimental period.

In this paper we present the Building RC Modeling (BRCM) Matlab Toolbox that facilitates the fast generation of MPC usable building models from basic geometry, construction and building systems data. Apart from the model generation, the Toolbox also provides parameterized functions for generating the (potentially time-varying) costs and constraints as required by MPC. The used modeling principles base on the already successfully validated approaches [10] and [12]. The input data are retrieved from a set of well defined files which to a large part can also be generated directly from EnergyPlus input data files. The Toolbox further provides basic functionality for the three-dimensional display of the target building's geometry, and for the simulation of a discrete-time representation of the model in open- and closed-loop modes. Figure 1 shows such a visualization of the second floor of the office building controlled in the OptiControl-II project. The BRCM Toolbox is open source and can be used under the GPLv3 license. The detailed documentation and installation instructions can be found on the BRCM website[2].

The rest of the paper is organized as follows. In Section II we introduce the modeling concept upon which the Toolbox is based. Section III describes the Toolbox' functionality, software interfaces and implementation. In Section IV a small case study is shown in which the Toolbox was used to generate a one-zone model that was subsequently compared

[1]Automatic Control Laboratory, ETH Zurich, www.control.ee.ethz.ch
[2]Synergy BTC AG, Bern, Switzerland, www.synergy.ch

[1]A widely used building simulation software [11].
[2] www.brcm.ethz.ch

to an EnergyPlus model. In Section V, the support for the modeling approach is reviewed and it is discussed how the Toolbox compares to other modeling frameworks. In Sections VI and VII finally, planned extensions are mentioned and conclusions are drawn.

II. Modeling Concept

We first introduce the MPC optimization problem and the resulting requirements that a model has to satisfy such that it can be used in an MPC controller. In the second part we outline on a conceptual level the modeling approach used in the Toolbox. In the third part, we describe the underlying modeling assumptions of the thermal model and the currently supported EHF models.

A. Building Models for MPC

While some also consider highly nonlinear building models for MPC (e.g. [3]), most use formulations that can be cast into a convex optimization problem. The latter approach is also the focus of the BRCM Toolbox. Whenever MPC for buildings is mentioned in the rest of the paper, specifically the latter approach is meant. We assume that linear costs are employed which is reasonable since the cost function typically represents the consumed energy or a linear function of it (e.g. monetary costs). Using x_k to denote the states (temperatures of rooms or wall/floor/ceiling layers), u_k the inputs (e.g. heating power or blinds position) and v_k the predicted disturbances (e.g. solar radiation or ambient temperature) at prediction time step k, the resulting MPC problem looks as follows

$$\min_{u_0 \dots u_{N-1}} \sum_{k=0}^{N-1} c_k^T u_k \tag{1a}$$

$$\text{subj. to} \quad x_{k+1} = A x_k + B_u u_k + B_v v_k + \dots \sum_{i=1}^{n_u} (B_{vu,i} v_k + B_{xu,i} x_k) u_{k,i} \tag{1b}$$

$$F_{x,k} x_k + F_{u,k} u_k + F_{v,k} v_k \leq f_k \quad \forall k = 0, 1, \dots, N-1 \tag{1c}$$

$$x_0 = x. \tag{1d}$$

In this, x denotes the estimated state of the system at the beginning of the MPC horizon of length N, n_u the number of control inputs and $u_{k,i}$ the i-th element of u_k. Furthermore, A, B_u, B_v and $B_{vu,i}$, $B_{vx,i}$ $i = 1, \dots, n_u$ define the system dynamics (1b). Finally, $F_{x,k}$, $F_{u,k}$, $F_{v,k}$, f_k and c_k denote the constraint matrices and vector as well as the cost vector at prediction time step k. The BRCM Toolbox is concerned with creating the system matrices and providing functions to generate $F_{x,k}$, $F_{u,k}$, $F_{v,k}$, f_k and c_k as a function of potentially time-varying parameters (e.g. minimum supply air temperature or electricity costs).

Note that the convexity of the MPC problem is usually not guaranteed for nonlinear system dynamics such as the bilinear model in (1b). These bilinearities appear in many simplified building models since they are usually necessary to model the product of temperature and mass flow rate as they appear for instance in ventilation models. In the rest of the paper, whenever bilinearity is mentioned, we specifically mean bilinearity in x and u or in v and u. Since usually the disturbances v are assumed to be perfectly predicted, the predictions $v_0, \dots, v_{N-1}$ can be directly plugged into (1b), resulting in a time-varying additive term and B_u matrix. However, the $x \cdot u$ bilinearities still remain. Many possibilities exist to solve the resulting nonlinear optimization problem. Most commonly, sequential linear programming is applied in which iteratively the bilinear x terms are substituted by the last calculated x trajectory and the linear problem is solved until convergence is achieved. This approach has been successfully applied in many cases, also because the system typically is only very mildly nonlinear. Hence, the bilinear form can be thought of as facilitating the linearization of an otherwise nonlinear system around trajectories $x_0, \dots, x_N$ and $v_0, \dots, v_N$ while providing enough modeling flexibility for most building MPC applications.

B. Stepwise Modular Modeling Concept

In [10] we introduced the distinction between the dynamic thermal model and static EHF models. It allows the separation of the rather generic modeling of the building's thermal dynamics from the much more building specific modeling of building systems and other external influences. This case-to-case variability is handled by the modular addition of parameterized EHF models for all building systems and other external influences present in the case at hand. The thermal model is derived from building geometry and construction data, while setting the EHF models parameters additionally requires building systems data. In Section III-B the input data required by the Toolbox is described in more detail.

The building's thermal model[3] as a function of EHFs $q(x(t), u(t), v(t))$ is given by

$$\dot{x}(t) = A_t x(t) + B_t q(x(t), u(t), v(t)). \tag{2}$$

The EHF models as functions of the states, control inputs and predictable disturbances,

$$q(x(t), u(t), v(t)) = A_q x(t) + B_{q,u} u(t) + B_{q,v} v(t) + \dots \sum_{i=1}^{n_u} (B_{q,vu,i} v(t) + B_{q,xu,i} x(t)) u_i(t), \tag{3}$$

[3] We use the subscript "t" to denote matrices of the thermal model (e.g. A_t) and subscript "q" for matrices of the EHF models

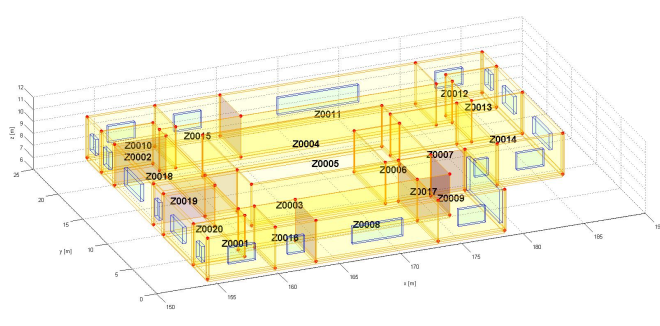


Fig. 1. Visualization of the geometry of the second floor of the OptiControl-II target building produced by the BRCM Toolbox.

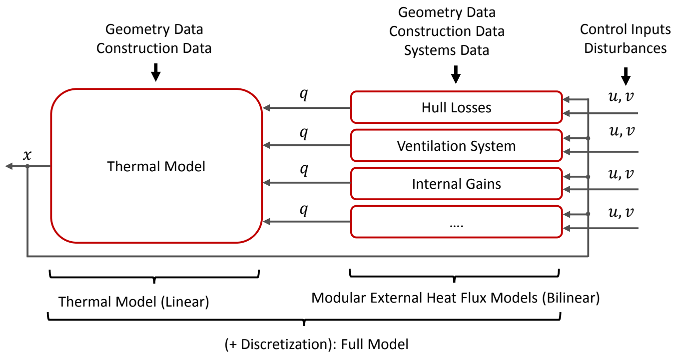


Fig. 2. Schematic description of the modular modeling approach. A linear thermal model is generated from construction and geometry data. To this a set (depending on the case at hand) of modular EHF models are added parameterized with geometry, construction and systems data. A discretization of the combined model yields the full model.

have in general to be modeled bilinearly as discussed above. Model (2) is then combined with (3) and subsequently discretized to obtain (1b), which in the following we will refer to as the *full model*. Figure 2 depicts this workflow schematically.

C. RC Modeling Approach

The core model generation uses the algorithms described in [10] which in turn are mainly based on the one-zone model [12]. In [10] we individually parameterized and connected multiple of these one-zone models to build a full building model and validated it against a complex building model in EnergyPlus. The approach makes the following main assumptions for the thermal model of the building: i) the air volume of each zone has uniform temperature; ii) temperatures within building elements vary only along the direction of the surface normal; iii) there is no conductive heat transfer between different building elements; iv) the temperature within a layer of a building element is constant; v) all model parameters are constant over time; vi) long-wave (i.e. thermal) radiation is considered in a combined convective heat transfer coefficient (this is likely to be changed in the future, see Section V).

With respect to the thermal building model, assumption i) and iv) directly lead to modeling the temperature (or thermal energy) of every zone's air mass and every building element's layer using one state. Assumption ii) leads to modeling the (conductive) heat transfer within a building element linearly proportional to the temperature difference between neighboring layers. Assumptions iii) and vi) imply that direct heat transfer among different building elements is neglected. Finally, based on assumptions i) and ii), heat transfer between zone air and surface building element layers is modeled to be proportional to the respective temperature difference.

Several boundary conditions are supported: *adiabatic*, *ambient* (convective heat transfer and solar radiation), *ground* or *prescribed temperature* (conductive heat transfer related to a disturbance input that models the ground or some other temperature). Additionally, it is possible to connect the first and the last layer of a building element to the same zone air state in order to approximate a situation in which many identical zones are adjacent to each other.

The Toolbox currently supports the following EHF models (for details we refer to the documentation on the webpage):
Building Hull: This EHF model considers convective heat transfer to all opaque facade parts and convective/conductive heat transfer through statically modeled windows to the zones (modeled by a U-value). Solar radiation onto the facades is modeled as disturbance inputs[4]. Solar heat gains to the opaque facade parts are modeled by an absorption coefficient. Thermal radiation exchange of the opaque facade parts is considered in the convective coefficients. Solar heat gains through windows are considered by scaling the incident solar radiation with a constant solar heat gain coefficient (SHGC) and the current blinds position (if present, modeled as a controllable gain between 0 and 1). The primary solar heat gains are distributed proportionally to their surface area among the innermost building element layers while the secondary heat gains are added to the zone air.
Internal Gains and Radiators: Internal gains and radiators are modeled as simple heat gains to the zones with the difference that internal gains are defined by a disturbance input while radiators are a controlled heat flux.
Ventilation: The supported ventilation systems can include a controllable air mass flow, a controllable energy recovery system, a controllable supply air heater and/or cooler and a controllable adiabatic return air cooler. Air flow paths, used to model situations where air supply and exhaust are not located in the same zone, can be specified. It is possible to constrain the supply air temperature.
Building Element Heat Flux: This EHF model can be used to model building systems such as floor heating, chilled ceilings or thermally activated building systems (TABS). These systems are modeled as simple controllable heat gains to the respective building element layers.

III. Toolbox

After the general introduction of the Toolbox in Section I and the conceptual modeling description in Section II, we describe the Toolbox from the viewpoint of specific aspects: Functionality, software interfaces and implementation. A usage example is given in Section IV.

A. Functionality

Loading of thermal model input data. For the thermal model, seven input data sheets with well-defined structures (see Section III-B) are parsed into data objects (see Section III-C for an overview of all the mentioned objects' classes). Supported file formats are .csv, .xls, .xlsx. Note that OpenOffice Calc documents can also be saved in Excel format, hence there is no dependence on the Excel software, even if one does not want to use .csv.
Programmatical manipulation of parameters. All parameters

[4] Note that this implies that those disturbance inputs have to be externally calculated from global solar radiation on a horizontal surface (which is typically what is available). The fact that this calculation is not internalized in the Toolbox is a major drawback. The next version of the Toolbox will incorporate those algorithms. See Section V.

in the data objects not related to geometry can be programmatically modified, for example for sensitivity studies. In particular the Toolbox supports easy, model-wide changes to key model parameters via the parameters object.
Saving of model data. The manipulated data objects can be saved back to disk in the same input data format.
Visualization. Based on the thermal model data, a 3D plot of the building can be generated, for instance for visual checking of the thermal model's geometrical correctness and completeness, see Figure 1.
Generation of thermal model. A thermal model object corresponding to (2) is automatically created from the data objects. This is based on the algorithms in [10].
Loading and generation of EHF models. For each required EHF model, a separate, specific data sheet is loaded that contains all necessary parameters and the corresponding model objects are generated.
Generation of full continuous-time model. Combination of the EHF models with the thermal model into the full continuous-time building model.
Discretization Discretization of the continuous-time full and/or thermal model.
Simulation. The full model can be simulated either in open-loop by specifying control input and disturbance trajectories or in closed-loop by providing a handle to a user-programmed controller function.
Parameter dependent generation of costs and constraints. The cost and constraint matrices can each be generated by a call to a function that assembles the individual cost contributions and constraints of every EHF model as a function of EHF model-specific parameters. This allows the generation of time-varying costs and constraints, for example resulting from time-varying electricity prices or occupancy dependent supply air temperature constraints, see Section IV.
EnergyPlus to BRCM input data files. Matlab based translation of the EnergyPlus input data file. Note that only thermal model input data parsing is supported, EHF models must be manually specified.

B. Interfaces

In this section we describe the four classic software interfaces, application programming interface (API), graphical user interface (GUI), output interface and input interface with an emphasis on the last.

The core of the Toolbox is a strong API through which all operations can be executed. Apart from the building visualization, a GUI was not implemented.

The output of the Toolbox is the discrete-time model (1b) and associated cost and constraint generating functions. For the sake of flexibility, the Toolbox does not save the output to a file in a specific format but is just concerned with creating a building model object containing all necessary information from which any required outputs can be flexibly generated.

Regarding the input interface, we distinguish the thermal model input data and the EHF model input data. In the rest of this section, the (significantly more extensive) thermal model input data is described. For the input data descriptions of all EHF models we refer to the documentation on the webpage. The thermal model input data was defined such that any number of thermal zones, floor plans including oblique walls, oblique floors/ceilings, windows and structural openings can be modeled. It consists of seven distinct input data files (.xls/.xlsx/.csv). Every file consists of one header row and an arbitrary number of data rows. Table I shows the first few lines of a zones data sheet example. All headers have two fields in common. A unique identifier field and a description field where an arbitrary string can be entered, for instance for creating legends in the simulation results plots. The other header fields are described below. The software supports for all not geometry related fields plus for selected geometry related fields entering of an algebraic expression instead of simple numerical values. The algebraic expressions are evaluated at the time of model generation and may contain parameter identifiers defined in the parameter data sheet.

1) Zones:
Area. Numerical. Zone floor area. *Volume*. Numerical. Zone volume. *Group*. (optional) Zone group identifier string. Can be used to address groups of zones.

2) Building Elements:
Construction. Identifier of a construction or a no-mass construction (no-mass constructions are modeled as a stateless thermal resistance, e.g. to model openings). Defines the construction type of the building element. *Adjacent A.* Zone identifier or special boundary condition identifier. Denotes the building element's boundary condition on side A. *Adjacent B.* Same as before but for side B. *Window Identifier.* (optional) Window identifier. Defines the building element's window type. *Area.* (optional if vertices are specified) Numerical or algebraic expression. Gross area of the building element, i.e. including window area. *Vertices.* (optional if area is specified, but no visualization possible if lacking) 3D numerical coordinates of the building element's corners in clock- or anticlockwise direction. Must lie in a plane.

3) Constructions:
Materials. Ordered list of material identifiers. Denotes the materials of the layers within the building element. By convention, the first entry corresponds to the layer at side A of the building element and the last entry to side B. *Thickness.* Numerical or algebraic expression. List of thicknesses of the layers in the same order as the materials list. *Convective Heat Coefficient Adjacent A.* Numerical or algebraic expression. Convective heat transfer coefficient on building element side A. *Convective Heat Coefficient Adjacent B.* Same for side B.

4) No-Mass Constructions:
U-Value. Numerical or algebraic expression. Heat transfer coefficient of a building element that is to be modeled statically, i.e. without states.

5) Materials:
Specific Heat Capacity. Numerical or algebraic expression. *Specific Thermal Resistance.* Numerical or algebraic expression. *Density.* Numerical or algebraic expression. *R-Value*. Numerical or algebraic expression. Can be used to enforce a stateless modeling of a layer (other fields must be empty).

6) Windows:
Glass Area. Numerical or algebraic expression. Area of the window glass. *Frame Area.* Numerical or algebraic expression. Area of the window frame. *U-Value.* Numerical or algebraic expression. Combined total heat transfer coefficient

TABLE I

FIRST FEW LINES OF A ZONES DATA SHEET EXAMPLE.

Identifier	Description	Area	Volume	Group
Z0001	Zone1 West	10	3	WestGroup
Z0002	Zone2 West	12.5	3	WestGroup
...	...	...	...	...

of frame and window including convective coefficients. *Solar Heat Gain Coefficient.* Numerical or algebraic expression. Coefficient scaling the global solar radiation on the outside of the window. Considers primary and secondary solar heat gains from window glass and frame.

7) Parameters:
Value. Numerical. Value of the parameter.

C. Implementation

The BRCM Toolbox has been coded in pure Matlab without employing any other toolboxes. The code currently encompasses around 8000 lines of code. Figure 3 shows the class structure with the two main classes Building and SimulationExperiment.

Of particular interest is the implementation of the EHF models. The following architecture ensures not only the provision of the EHF system matrices but also of functions generating corresponding constraints and costs and enables the addition of further EHF models by other parties. Any EHF model is an object of a class satisfying the following properties: i) it has been derived from a base EHF model class; ii) its constructor takes as input just the Building object and a path to an input data sheet that parameterizes the EHF model; iii) based on the input data sheet, the constructor generates the EHF model matrices and implements the abstract EHF model base class methods getConstraintsMatrices and getCostVector. Since the EHF model depends only on its "own" control and disturbance inputs, functions are provided in the EHF model base class that generate the full sized model, constraints matrices and cost vector given the identifiers of the full model's control and disturbance inputs.

If these conditions are satisfied, the Toolbox can iterate over all EHF model objects, call these functions and compile the results into the full model and the full model's constraints and costs matrices/vectors.

IV. CASE STUDY

In this section we compare a BRCM to an EnergyPlus model. The section should illustrate how the Toolbox can be used to generate a model and demonstrate its predictive capabilities in open-loop.

For simplicity, we chose a single south facade zone of the building shown in Figure 1. For a multi-zone case see [10]. This allowed a comparison between the BRCM and the EnergyPlus model on a more detailed basis. First the single zone was simulated for five days in EnergyPlus using weather data from April 1, 2010 in Basel. For this we used the convenient MLE+ toolbox [13] to couple Matlab and EnergyPlus by BCVTB [14]. The EnergyPlus model was initialized such that the whole building had an initial temperature of 23°C. In

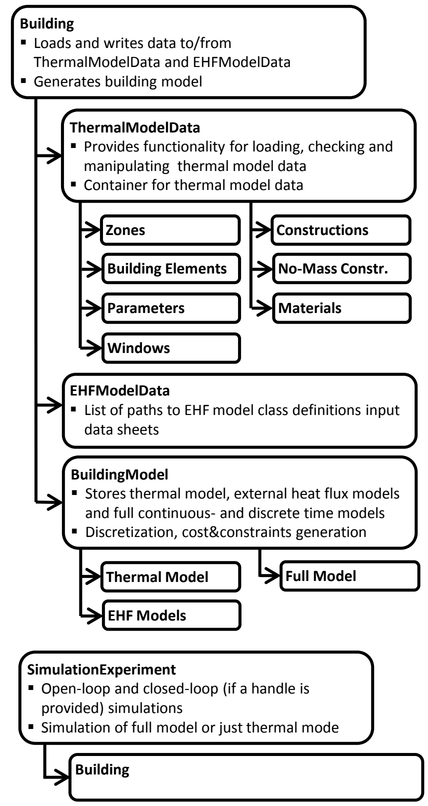


Fig. 3. Class structure of the BRCM Toolbox. An arrow indicates that an object of the class pointed to is included in the class the arrow originates from.

this way it was possible to set the same initial conditions in both models such that all discrepancies between the models stemmed from model differences and not different initial conditions. The EnergyPlus model (v7.0) used default algorithms for calculating heat transfer (TARP,ConductionTransferFunction). From the EnergyPlus simulation outputs the room and surface temperatures as well as the disturbance inputs to the BRCM model were extracted (i.e., the weather data). From the EnergyPlus input data file, we then generated the input data sheets for the BRCM thermal model and specified the EHF model related to the building hull heat fluxes. In the following, we show how the BRCM model was generated, simulated and compared to the resulting temperature trajectories.

To install the BRCM Toolbox, only one Matlab command has to be issued, assuming the Toolbox Manager [15] is installed.

```
tbxmanager install brcm;
```

First, a new building object was created. Then the thermal building model data was loaded and the EHF model to be included was declared. For simplicity we only considered a single EHF model.

```
B = Building;
B.loadThermalModelData(pathToThermalModelData);
B.declareEHFModel('EHFM_BuildingHull.m',pathToEHFMData,id);
```

The following commands programmatically change the field 'Value' of some model parameter 'parameterID' to 10 and save the resulting thermal model back to the input data format. The commands are shown just for illustration purposes; the changes have not been made to the model that was compared to EnergyPlus.

```
B.thermal_model_data.setValue('parameterID','Value',10);
B.writeThermalModelData(pathToModifiedThermalModelData);
```

The visualization command results in a single zone model version of Figure 1.

```
B.drawBuilding();
```

We then generated the continuous-time building model and discretized it.

```
B.generateBuildingModel();
B.building_model.discretize(discretizationTimestep);
```

For the open-loop simulation, we assumed that the control and disturbance input matrices U, V had already been extracted from EnergyPlus. We created a new simulation experiment object SimExp, set the initial condition and performed an open loop simulation with the control and disturbance input matrices.

```
SimExp = SimulationExperiment(B.building_model);
SimExp.setNumberOfSimulationTimeSteps(numberOfTimesteps);
SimExp.setInitialState(23*ones(number_states,1))
SimExp.simulateBuildingModel('inputTrajectory',U,V);
```

In Figure 4(a) we show the ambient air temperature and the solar global radiation on the facade. In Figure 4(b), the room temperatures of both models are shown. Finally, Figure 4(c) shows the differences (RC-EP) of the building element surface temperatures and the room temperature difference. The surface temperatures varied in a same range as the room temperature. Over the five day period, the zone air temperature difference was within 0.5 °C while the wall building element temperatures mostly agreed well too except for times with high solar radiation. This is due modeling the thermal radiation indirectly by a modified convective heat transfer coefficient, see Section V for a discussion of this point.

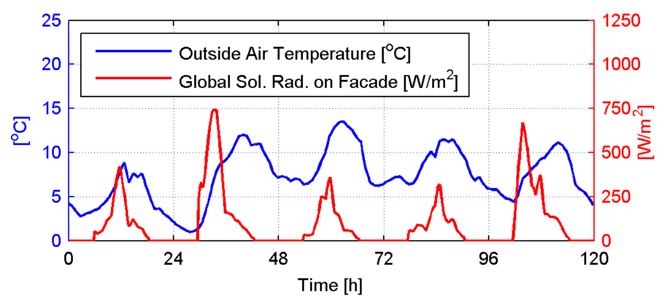
For a model with control inputs, the following commands would generate the current cost and constraints matrices based on the current constraint and cost parameters conP, cosP, respectively. Which parameters are necessary naturally depend on the included EHF models. Again this command is just shown for illustration purposes.

$[F_{x,k}, F_{u,k}, F_{v,k}, f_k]$ = B.building_model.getConstraintsMatrices(conP);
c_k = B.building_model.getCostVector(cosP);

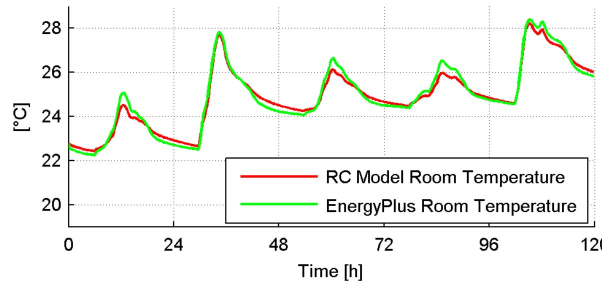
V. Discussion

The goal of the BRCM Toolbox is to generate models and according costs and constraints for MPC, i.e. in a (bi-)linear form. Much of the justification for using the Toolbox depends on the gathered support for its modeling approach. In the basic single-zone model developed in [12], the modeling approach was compared to Trnsys. The modeling algorithms from [6] on which the Toolbox is directly based have been validated against EnergyPlus. In the OptiControl-II project [8], a such generated model was successfully used in the MPC on the real building and also validated by comparing predicted and measured temperatures in step experiments. Finally, in Section IV a small single-zone model was developed and again compared against EnergyPlus.

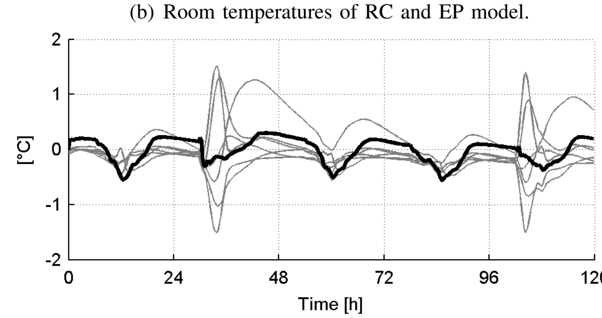
Even though the goal of the Toolbox is not to compete with the large number of sophisticated building simulation frameworks, we think it is important to know the differences in modeling. Here we compare the modeling approach of the Toolbox to EnergyPlus. Not considering differences in the modeling of building systems, the following are the main differences (assuming default EnergyPlus settings): i) constant model parameters; ii) thermal radiation within zones considered in a combined convective/radiation coefficient; iii) windows modeled statically and having to rely on pre-calculated incident solar radiation on building facade ele-



(a) RC simulation inputs. Outside temperature and global solar radiation on the facade.



(b) Room temperatures of RC and EP model.



(c) RC-EP building element surface temperature differences in grey. RC-EP room temperature difference in bold.

Fig. 4. Comparison of a BRCM to a EnergyPlus model.

ments and windows.

Regarding i): EnergyPlus uses specialized algorithms to compute convective coefficients based on the temperature distribution in a zone. Detailed simulations in which we fixed these coefficients in EnergyPlus and used the same in the BRCM model showed that this approximation is the major source of the model discrepancies as shown for instance in Figure 4(c).

Difference ii) was also shown to be important. Modeling combined convective/radiation coefficients is a well accepted modeling simplification, but may be not sufficiently accurate. Radiative heat exchange among building elements, although highly nonlinear, can be well linearized and has been as a test (the current version does not support this) included in the linear thermal dynamics model. A comparison to EnergyPlus showed another improvement of the prediction capability of the model.

Regarding iii): EnergyPlus employs very sophisticated window models. In the Toolbox, this was approximated by constant U-Values and SHGC-Values and shading devices that are modeled to scale the solar heat gain. To solve the MPC problem (1), the disturbances $v_0, \cdots, v_{N-1}$ have to be predicted. Currently, the Toolbox assumes that the solar radiation values onto the facades are available as disturbances, even though typically only global solar radiation on a horizontal (or perpendicular to the sun) surface is forecasted[5]. The computation of the facade values is not trivial and currently not supported. However, within the OptiControl-II project Matlab algorithms for this have been developed and checked using measured radiation data from the real building. These algorithms are planned to be included in the Toolbox.

The most important underlying principle of the Toolbox is the choice to model the building thermal dynamics from physical first principles. The most prominent reason for this choice is the fact that due to time and building usage constraints the effort of identifying multi-input multi-output building models, which are desirable for the integrated control of office buildings' HVAC and blinds systems, may well be prohibitive or due to limited excitation even impossible in practice. Further drawbacks include the missing physical interpretation of an identified model and difficulties in handling changes in the buildings such as added or removed walls etc. The major downside with physical based modeling is that materials data often is not easily available and guesses have to be made. This downside is in our opinion heavily outweighed by the advantages.

VI. Future Work

Aside from directly considering thermal radiation, including the functionality for the solar calculations and further validation studies, several additional works are planned: Including model reduction routines, adding the possibility to choose how many states per layer are modeled and implementing input data interfaces also to further sources other than EnergyPlus.

VII. Conclusions

In this paper we presented a new Matlab Toolbox capable of generating (bi-)linear building models as well as costs and constraints for MPC. To our knowledge, this is the first framework aiming at generating full building models for MPC. The underlying modeling principles have been validated in simulation by comparing against EnergyPlus and Trnsys and in practice by using a such generated model in an actual model predictive controller.

In summary, the Toolbox has been thoroughly tested and all documented functionality works well. The currently most restrictive limitation - relying on externally calculated solar radiations - will soon be removed.

VIII. Acknowledgment

We would like to thank Carina Sagerschnig (Gruner AG, Switzerland) and Markus Gwerder (Siemens Switzerland Ltd.) for their great support. This work was financially supported by Siemens Switzerland Ltd. and swisselectric research.

References

[1] M. Gwerder and J. Tödtli, "Predictive control for integrated room automation," in *Proc. of the Clima - RHEVA World Congress*, 2005.

[2] F. Oldewurtel, *Stochastic model predictive control for energy efficient building climate control*. PhD thesis, Eidgenössische Technische Hochschule ETH Zürich, Nr. 19908, 2011.

[3] Y. Ma, F. Borrelli, B. Hencey, B. Coffey, S. Bengea, and P. Haves, "Model predictive control for the operation of building cooling systems," *IEEE Trans. on Control Systems Technology*, vol. 20, no. 3, pp. 796–803, 2012.

[4] Y. Ma, S. Richter, and F. Borrelli, "DMPC for Building Temperature Regulation," *in: Control and Optimization with Differential-Algebraic Constraints*, vol. 23, p. 293, 2012.

[5] J. Siroky, F. Oldewurtel, J. Cigler, and S. Privara, "Experimental analysis of model predictive control for an energy efficient building heating system," *Applied Energy*, vol. 88(9), pp. 3079–3087, 2011.

[6] D. Sturzenegger, D. Gyalistras, M. Gwerder, C. Sagerschnig, M. Morari, and R. S. Smith, "Model Predictive Control of a Swiss Office Building," in *11th REHVA World Congress Clima 2013*, 2013.

[7] A. Aswani, N. Master, J. Taneja, D. Culler, and C. Tomlin, "Reducing transient and steady state electricity consumption in HVAC using learning-based model predictive control," *Proceedings of the IEEE*, vol. 100, no. 1, pp. 240–253, 2012.

[8] M. Gwerder, D. Gyalistras, C. Sagerschnig, R. Smith, and D. Sturzenegger, "Final report: Use of weather and occupancy forecasts for optimal building climate control Part II: Demonstration (OptiControl-II)." http://www.opticontrol.ethz.ch/Lit/Gwer_13_Rep-OptiCtrl2FinalRep.pdf. ETH Zurich, 2013.

[9] J. Cigler, D. Gyalistras, V.-N. Tiet, and L. Ferkel, "Beyond theory: the challenge of implementing Model Predictive Control in buildings," in *11th REHVA World Congress Clima 2013*, 2013.

[10] D. Sturzenegger, D. Gyalistras, M. Morari, and R. Smith, "Semi-automated modular modeling of buildings for model predictive control," *BuildSys'12 Proc. of the Fourth ACM Workshop on Embedded Sensing Systems for Energy-Efficiency in Buildings*, pp. 99–106, 2012.

[11] *EnergyPlus.* http://apps1.eere.energy.gov/buildings/energyplus

[12] B. Lehmann, D. Gyalistras, M. Gwerder, K. Wirth, and S. Carl, "Intermediate complexity model for model predictive control of integrated room automation," *Energy and Buildings*, vol. 58, pp. 250–262, 2013.

[13] T. X. Nghiem, "MLE+: a Matlab-EnergyPlus Co-simulation Interface." http://www.seas.upenn.edu/~nghiem/mleplus.html.

[14] *BCVTB.* http://simulationresearch.lbl.gov/bcvtb

[15] *Toolbox Manager.* http://www.tbxmanager.com

[5] The other disturbances are more easily predicted. Typically, a forecast of the ambient temperature is readily available and internal gains profiles can be assumed (potentially based on electricity measurements).