Modeling Environment for Model Predictive Control of Buildings

T. Zakula\textsuperscript{a,}\textsuperscript{*}, L. Norford\textsuperscript{b}, P.R. Armstrong\textsuperscript{c}

\textsuperscript{a}University of Zagreb, FAMENA, I.Lucica 5, 10000 Zagreb, Croatia
\textsuperscript{b}Massachusetts Institute of Technology, 77 Massachusetts Avenue, Cambridge, 02139 MA, USA
\textsuperscript{c}Masdar Institute of Science and Technology, Masdar City, Abu Dhabi, UAE

Abstract

Model predictive control (MPC) is an advanced control that can be used for dynamic optimization of HVAC equipment. Although the benefits of this technology have been shown in numerous research papers, currently there is no commercially or publicly available software that allows the analysis of building systems that employ MPC. The lack of detailed and robust tools is preventing more accurate analysis of this technology and the identification of factors that influence its energy saving potential. The modeling environment (ME) presented here is a simulation tool for buildings that employ MPC. It enables a systematic study of primary factors influencing dynamic controls and the savings potential for a given building. The ME is highly modular to enable easy future expansion, and sufficiently fast and robust for implementation in a real building. It uses two commercially available computer programs, with no need for source code modifications or complex connections between programs. A simplified building model is used during the optimization, whereas a more complex building model is used after the optimization. It is shown that a simplified building model can adequately replace a more complex model, resulting in significantly shorter computational times for optimization than those found in the literature.

Keywords: model predictive control, simulation tool, energy efficiency

1. Introduction

Model predictive control (MPC) is an optimal control that uses a dynamic system model and predictions of future events to optimize the objective function (e.g. energy consumption or cost). MPC is already used in some process industries, and is becoming an increasingly popular research topic in buildings, demonstrating the benefits for building energy consumption and electricity cost. Based on weather and load predictions, MPC enables energy efficient strategies through the optimization of heating, ventilation and air conditioning (HVAC) system operation, while ensuring thermal comfort for occupants. For example, it can predict optimal shifting of cooling loads to night time, when...
the outside temperatures are lower and, therefore, the efficiency of cooling equipment is higher. Furthermore, it can result in lower cooling cost in case of utility rates that favor night operation, as well as reduction in peak loads.

The benefits of MPC have been demonstrated, mainly for variable-air-volume (VAV) systems, in numerous papers found in the literature. For a single zone building, Braun (5) showed 10–50% energy cost savings and 10–35% peak power reduction for the VAV system operated under MPC. Even without special time-of-day utility rates, the system with MPC resulted in 35% savings in the operating cost, mainly due to "free" night cooling. In subsequent work Lee and Braun (22) applied the advanced control on the ERS building at the Iowa Energy Center, reducing the peak cooling power demand by 25–45%. Krarti et al. (19) developed a modeling environment for the optimal control of an ice-storage system, used by Henze et al. (15) to analyze the MPC potential for a building with active thermal storage. They investigated different lengths of the planning horizon, where the planning horizon represented the time interval over which the cost function was evaluated. Results showed that the planning horizon on the order of 24 hours is only marginally sub-optimal compared to the horizon over a simulation period of one week. Krarti and Henze (20) showed 10–20% reduction in the operating cost by applying MPC for a prototypical three-story office building with active and passive storage and VAV system. The analysis of short-term weather forecasting models suggested that the bin model (which uses the observations from the previous 30 and 60 days) had the best prediction accuracy. The use of this model resulted in marginally different cost savings compared to the case with perfect weather knowledge. In the more recent study, Oldewurtel et al. (24) showed the benefits of Stochastic Model Predictive Control (SMPC) strategy that accounts for the uncertainty due to the use of weather predictions. For a large number of different cases varying in the building type, HVAC system, and weather conditions they demonstrated a significant energy savings potential of SMPC relative to the rule-based control. Alvarez et al. (1) combined Practical Non-Linear Model Predictive Control (PNMPC) and Lagrange dual optimization algorithm to find the optimal PMV index and maximize the thermal comfort in a multi-zone building with fan-coil units and a solar cooling system. In the analysis of a building with fan-coil units, Gruber et al. (13) reported between 5-46% energy savings when benchmark feed-back controllers were replaced by a model-based controller. They also found that, compared to more complex model-based controllers, simplified controllers for both temperature and CO2 control do not compromise energy savings to any larger extent. Armstrong et al. (2) analyzed "low-lift cooling," the technology that utilizes radiant cooling, an thermal active storage and MPC. Based on scoping studies for five different US climates, the advanced control strategy resulted in 30–70% energy savings compared to a typical VAV system. The savings were achieved mainly through fan energy reductions and shifting of the cooling loads toward night time, taking advantage of lower night temperatures. Gayeski et al. (12) continued the low-lift cooling technology analysis on a small-scale pilot project. Building mass acted as a passive thermal storage through night precooling and the use of thermally activated building surface (TABS). The results for a typical summer week in Atlanta and Phoenix showed sensible cooling savings of 25 and 19% respectively relative to the variable refrigerant flow (VRF) system.

Only a few papers have given detailed descriptions of the tools used to simulate a building with MPC. Krarti and Henze (20) provided an in-depth overview of a simulation model in which EnergyPlus was modified and integrated with the optimization
software GenOpt. The additional model was done in TRNSYS, using a version of the TRNSYS source code not commercially available (14). Optimizing a day with 24 hourly setpoints took 1–4 hours for the Nelder-Mead simplex method and 8–29 hours with the OptQuest (population-based scatter search) method. Spindler and Norford (27) and May-Ostendorp et al. (23) showed that MPC can also be successful in optimizing a mixed-mode building behavior. While Spindler and Norford (27) used the data-driven, inverse model trained on a real building, May-Ostendorp et al. (23) employed the combination of EnergyPlus and MATLAB environments. Using the particle swarm optimization, May-Ostendorp et al. (23) optimized window operation in a mixed mode building with a 24-hour planning horizon and a two-hour optimization block. This resulted in 12 optimization variables of binary window decisions (window in position 0 or 1). The reported simulation time for 11 weeks in summer was 12 hours. Corbin et al. (7) described a framework for MPC that combines EnergyPlus and MATLAB and uses the particle swarm optimization. The algorithm can be used for MPC of different building systems, which was shown for a VAV system and for a building with TABS. In the first example, having 14 daily temperature setpoints as optimization variables resulted in the simulation time of 26 clock hours to simulate one week. In the second example, it took one day of clock time to simulate one day for a building with 11 thermal zones and 12 optimization variables. Coffey et al. (6) developed a software framework for MPC that combines GenOpt and SimCon with any building energy simulation program that can read and write into a text file. The connection between SimCon and the energy modeling software TRNSYS was enabled through the Building Control Virtual Test Bed (29). To optimize one day using MPC, the reported computational time was three nights.

Although the benefits of MPC have been demonstrated in numerous research papers, important challenges that still remain are the lack of tools for the system analysis and practical challenges facing real building implementation. Findings in the literature on potential energy and cost savings are highly dependent on a variety of factors, such as the building type, internal load, climate, equipment characteristics, and controls. The use of a computer model allows for a systematic study of primary factors influencing the dynamic control and savings potential for an individual building. However, currently there are no commercially or public available tools for this type of analysis. Of the few papers in the literature that give detailed descriptions of the tools used to simulate building with MPC, most require modification of existing building simulation programs, which has been shown as challenging. One example is a severe issue with initialization of variables in building software EnergyPlus and TRNSYS, as mentioned in more detail later in this paper. Furthermore, the reported computation times are not practical for implementation in real buildings where the optimization usually needs to be repeated each hour due to uncertainties in load and weather predictions. The implementation in real buildings is somewhat inhibited by control complexity compared to conventional systems. Examples are found in the literature of simplified control strategies that would result in a near-optimal control (16), but those strategies are still obtained by using more detailed computer models.

The modeling environment (ME) presented here simulates the performance of a building in which HVAC systems are operated using MPC. It can be set to optimize a variety of HVAC systems and optimization objectives, using different planning horizons (the time interval over which the objective function is evaluated) and execution horizons (the time interval over which the control strategy is applied). The ME does not require any
modification of existing building programs, only the common connection between Matlab and TRNSYS (using TRNSYS Type 155). To reduce computational time, the ME uses two building models of different complexities, where the simplified model is used in the optimization, and TRNSYS only for post optimization. This avoids complex connections between different programs and results in significantly shorter computational time, making the ME robust and suitable for implementation in real buildings.

2. Model description

Here we describe the modeling environment (ME) for the example of a cooling system controlled to minimize electricity consumption. The example will include cooling with thermally activated building surfaces (TABS) and a variable-air-volume (VAV) system. Different objectives, such as the total cost of electricity/fuel for cooling/heating, can be achieved by modifying the objective function.

The basic scheme of minimizing a building’s electricity consumption is shown in Figure 1. With the planning horizon of 24 hours, the optimization algorithm is executed day-by-day, making a first guess of cooling rates in each hour (24-variable optimization). The optimization variable for the VAV system is a sensible cooling rate imposed on the room, whereas the optimization variable for cooling with TABS is a chiller cooling rate, different than the instantaneous cooling rate imposed on the room due to the TABS thermal lag. The objective function that needs to be minimized is a sum of the total daily electricity for cooling, electricity for transport and the temperature penalty. Introducing the temperature penalty is an implicit way of ensuring thermal comfort for the occupants.

To determine whether the proposed strategy leads to comfortable building conditions, the ME incorporates a building model capable of predicting transient thermal response of the occupied zones. Thermal comfort is influenced by both air and surface temperatures, which becomes especially relevant for cooling/heating with TABS. Compared to all-air systems, the benefit of cooling with TABS is that people feel equally comfortable at slightly higher air temperatures due to large, cold surfaces. To account for the combined effect of air and surface temperatures, the operative temperature, rather than the air temperature, was chosen as the controlled temperature. In a real building, where surface temperature measurements are usually not available, a more practical solution would be to use the room air temperature as the control variable, but setting the temperature setpoint 1–2°C higher when cooling with TABS than when cooling with the VAV system.

After optimization using the simplified building model, the optimal control settings are passed to either a real or a virtual building. Parameters that will be passed depend on the type of cooling system; for example, the parameter used as the input for the building with the VAV system is the volumetric flow rate of air (assuming constant supply temperature), and the input for the building with TABS is the water supply temperature (assuming constant mass flow rate).

2.1. Building model

When predicting a control strategy that minimizes the cost function (e.g. the price of electricity or the total electricity consumption) it is necessary to capture a building’s thermal and hygric response. The thermal response is used to ensure thermal comfort for the occupants, while the hygric response becomes especially relevant for cases in which a
room’s humidity needs to be tightly controlled. For example, if the humidity of a room
cooled with thermally activated building surfaces (TABS) is not controlled, condensation
can occur due to cold surfaces. The intention here was to develop a MPC analysis tool
that can be widely used, and be fast enough for implementation in a real building. Due
to uncertainty in weather and internal gain predictions, the optimization in real buildings
usually needs to be repeated more frequently (e.g. each hour), putting a relatively strict
constraint on the computational time. This influenced the selection of a building model;
only commercially available tools that give a fast building response were considered.

Many commercially available Building Energy Simulation Programs (BESP) offer
more or less detailed simulation of a building’s dynamic behavior. There are numerous
programs for whole building energy simulation such as DOE-2, DesignAdvisor, eQUEST,
CAMSOl, SPARK, TARP, Modelica, etc. A complete list and overview of a program
capabilities can be found in the Building Energy Software Tools Directory (EERE).
Most of BESPs are focused on thermal calculations, while analyzing moisture transfer
with simplified methods that do not account for a change of material properties with
variations in temperature and humidity, nor do they account for a change in the moisture
buffer effect due to a temperature gradient in a wall. The simplified method often used to
describe moisture transfer between the air and solids is the effective moisture penetration
depth (EMPD) theory proposed by Kerestecioglu et al. (17). Due to a lack of programs
that combine airflow, heat transfer and moisture transfer processes in buildings, Annex
41 of the International Energy Agency (IEA) had the goal of stimulating the development
of information and analytical tools. Rode and Woloszyn (25) gave a detailed overview of
the IEA project, description of common exercises, advances in simulation programs and
papers published on the topic.

Two widely used programs developed to capture a building’s transient behavior are
EnergyPlus (8) and TRNSYS (Transient System Simulation Program, Klein et al. 2010).
These programs are often used in academic research since they enable a detailed analysis
of complex building systems. Numerous research papers can be found on their use for
variety of building analyses, with TRNSYS more often used in Europe and EnergyPlus
in the U.S. TRNSYS has been chosen for the MPC modeling environment as a compre-
hensive and modular simulation tool that is somewhat more suitable for building control
system analysis. EnergyPlus was not originally designed for a detailed analysis of building
control systems, but can be linked to programs more suitable for system controls, such as
MATLAB or Simulink. The connection can be done through Building Controls Virtual
Test Bed (BCVTB) developed by Lawrence Berkeley National Laboratory. However,
both TRNSYS and EnergyPlus proved to be challenging for the MPC application due
to their inability to explicitly initialize all variables at the beginning of a new planning
horizon period. For example, a user can explicitly define the initial room air temperature
before a simulation starts, and that temperature is also assumed to be the initial wall
temperature. Since the thermal mass temperature can be significantly different than the
room temperature in the building with TABS, the inability to explicitly define different
wall initial conditions represents a severe issue.

Both the issue of computational speed and especially the issue of initialization of
variables were motives to consider the use of an alternative model for the optimization
algorithm. The chosen alternative model is the inverse model based on a comprehensive
room transfer function developed by Seem (26). Seem (26) showed that the transfer
functions used to describe transient changes for individual surfaces can also be used
to describe transient changes of a whole room. In the derived comprehensive room
transfer function (CRTF), a heat flux for a room was calculated as a weighted sum
of past heat rates and instantaneous and past temperatures. The number of past terms
depends on the "heaviness" of the mass, where more past terms are relevant for a heavier
construction. Armstrong et al. (3) showed that the CRTF coefficients for heat flux
response are generally not best for temperature response (iCRTF). In their proposed
model, a temperature is a weighted sum of past zone temperatures and current and past
ambient temperatures and heat rates:

\[ T_z = \sum_{t=1}^{n} a^t T_z^t + \sum_{t=0}^{n} b^t T_z^t + \sum_{t=0}^{n} c^t q^t \]  

(1)

where the coefficients \( a, b \) and \( c \) must satisfy the Fourier’s law of conduction in steady
state and therefore, the constraint for the inverse model coefficients is:

\[ 1 - \sum_{t=0}^{n} a^t = \sum_{t=0}^{n} b^t \]  

(2)

Gayeski (11) showed that the inverse model can serve for a fast and reliable imple-
mentation of MPC in a real building. He used the linear regression on a few weeks of
training data to find the model coefficients, and then used the inverse model to predict
room temperatures in the 24-hour-ahead TABS cooling optimization. Gayeski also ana-
lyzed the dependence of prediction accuracy on number of past terms for a specific case.
He showed that a higher number of terms does not necessarily give better predictions
due to the strong effect of system noise on parameters of high-order identified models.

While the simplified, inverse building model is used for the optimization, a more
detailed building model in TRNSYS is used to find inverse model coefficients and validate
the inverse model. Moreover, it is employed as a virtual building to give a more accurate
building response on optimal control after the optimization.

2.1.1. TRNSYS model

A room cooled with the VAV system and parallel TABS (cooling through floor slab)
has been modeled in TRNSYS. The modeled room is the experimental room located at
Massachusetts Institute of Technology (MIT), USA, and it was chosen since experimen-
tal measurements for a typical summer week in Atlanta (11) could be used to validate
the accuracy of the TRNSYS model. The room is divided into the climate room and
test room, both adjacent to the larger laboratory room. The walls are made of two 16
mm gypsum layers, with 110 mm of a polyisocyanurate foam placed in between. There
are three double-pane windows between the test room and the climate room. The test
room floor has PEX pipes embedded into the commercially available subfloor system
and is covered with three layers of concrete pavers. Details on the room construction are
given in the Appendix. The pipes can be used for hydronic sensible cooling or heating.
The room has an additional indoor unit (VRF system) for direct heating, cooling and
dehumidification. Both systems are served by the common outdoor unit, which is placed
in the climate room so that it captures changes in the chiller efficiency with respect to
ambient conditions. The test room is also equipped with lights and heat sources that can
simulate internal convective and radiative heat gains for a typical office building. The
climate room temperature is controlled by a separate air heating and cooling system, allowing for testing different climate conditions. The experimental measurements were not performed for the hygric response of the MIT’s experimental room. Therefore, parameters used in the TRNSYS buffer storage model for hygric response are those given in the TRNSYS example for a typical office. The office is approximately of the same volume as MIT’s experimental room, with walls made of 12 mm gypsum and 100 mm mineral wool and the floor and ceiling made of concrete.

Experimental measurements that were inputs to the TRNSYS model for model validation are the climate room temperature, laboratory room temperature, floor temperature below TABS, internal loads, supply water temperature, and water mass flow rate. Cases were run in TRNSYS with time steps of 1, 15, and 30 minutes to ensure that the results are not affected by our choice of time step. The relative RMS error for predicted air temperature response, return water temperature, floor temperature and TABS cooling rate (Table 1) show good agreement with the measured data. The prediction errors were calculated as:

\[ \text{Relative RMSE} = \text{mean} \left( \frac{X_{\text{measurements}} - X_{\text{TRNSYS}}}{\max(X_{\text{measurements}}) - \min(X_{\text{measurements}})} \right)^2 \]  

(3)

<table>
<thead>
<tr>
<th></th>
<th>( T_z )</th>
<th>( T_{w,\text{return}} )</th>
<th>( T_{\text{floor}} )</th>
<th>( Q_{\text{TABS}} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>Relative RMSE</td>
<td>0.0512</td>
<td>0.0040</td>
<td>0.0339</td>
<td>0.0256</td>
</tr>
</tbody>
</table>

2.1.2. Inverse model for thermal response

Here we show how Equation 1 can be modified to describe the thermal response of a single zone cooled by the VAV system and parallel TABS. The equation is used to calculate the air temperature \( T_z \), operative temperature \( T_o \), floor temperature \( T_{\text{floor}} \) and water return temperature (from TABS) \( T_{w,\text{return}} \). The VAV system sensible cooling represents an instantaneous convective cooling rate, whereas the internal loads have a certain convective-to-radiative heat transfer split. Therefore, to improve the inverse model predictions in cases when the VAV system and internal loads are acting together, the third term in Equation 1 is split into a convective and radiative term. Cooling with the TABS is also treated as a separate term, since it is accounted for in the instantaneous chiller cooling rate, different than the instantaneous zone’s cooling rate due to the thermal lag.

Modified equations are:

\[ T_z = \sum_{t=1}^{n} a_{z}^{t} T_{z}^{t} + \sum_{t=0}^{n} b_{z}^{t} T_{adj}^{t} + \sum_{t=0}^{n} c_{z}^{t} T_{x}^{t} + \sum_{t=0}^{n} d_{z}^{t} Q_{TABS}^{t} + \sum_{t=0}^{n} e_{z}^{t} Q_{\text{conv}}^{t} + \sum_{t=0}^{n} f_{z}^{t} Q_{\text{rad}}^{t} \]  

(4)
\[ T_0 = \sum_{t=1}^{n} a_t^t T_0^t + \sum_{t=0}^{n} b_t^t T_{\text{adj}}^t + \sum_{t=0}^{n} c_t^t T_x^t + \sum_{t=0}^{n} d_t^t Q_{\text{TABS}}^t + \sum_{t=0}^{n} e_t^t Q_{\text{conv}}^t + \sum_{t=0}^{n} f_t^t Q_{\text{rad}}^t \]  

(5)

\[ T_{\text{floor}} = \sum_{t=1}^{n} a_f^t T_{\text{floor}}^t + \sum_{t=0}^{n} b_f^t T_{\text{adj}}^t + \sum_{t=0}^{n} c_f^t T_x^t + \sum_{t=0}^{n} d_f^t Q_{\text{TABS}}^t + \sum_{t=0}^{n} e_f^t Q_{\text{conv}}^t + \sum_{t=0}^{n} f_f^t Q_{\text{rad}}^t \]  

(6)

where \( Q_{\text{rad}} \) accounts for 50% of the internal loads from people, equipment and lights, and \( Q_{\text{conv}} \) for the other 50% of the internal loads and 100% of the additional cooling/heating. \( T_{\text{adj}} \) is the temperature of adjacent rooms. The cooling rate of the chiller is calculated as:

\[ Q_{\text{TABS}} = m_w c_w (T_{\text{w,return}} - T_{\text{w,supply}}) \]  

(8)

The coefficients \((a, b, c, d, e, f)\) can be found from the linear regression on training data created by the TRNSYS model, or from real building measurements.

If the room has only TABS cooling, without the parallel VAV system, Equations 4–6 can be simplified by combining internal loads into one term, and calculating the zone temperature, operative temperature and floor temperature as:

\[ T = \sum_{t=1}^{n} a_t^t T^t + \sum_{t=0}^{n} b_t^t T_{\text{adj}}^t + \sum_{t=0}^{n} c_t^t T_x^t + \sum_{t=0}^{n} d_t^t Q_{\text{TABS}}^t + \sum_{t=0}^{n} e_t^t Q_t^t \]  

(9)

To calculate the zone temperatures using Equations 4–6, the cooling rates delivered to the room by the VAV system must be known. However, the variables that are known in real buildings are the supply airflow rate and supply temperature, while the amount of delivered cooling depends on the zone temperature. To solve this coupling between the air temperature and cooling rates, the iteration loop shown in Figure 2 has been added to the inverse model. Although this addition somewhat increases the computational time, the inverse model is still significantly faster than the TRNSYS model. For example, to predict the zone’s thermal response for one week, it took about 0.005 seconds of real time for the inverse model without the additional interpolation loop, 0.5 seconds with the additional interpolation loop and 2-3 seconds for the TRNSYS model, all on an i7-2600 @ 3.4 GHz processor.

### 2.1.3. Validation of inverse model for thermal response

The TRNSYS model was used to find the coefficients \((a, b, c, d, e, f)\) from the linear regression on training data (Figure 3) and to create two additional data sets for inverse model validation. The first validation set (Figure 4a) represents the case with VAV system cooling, while in the second validation set (Figure 4b) both the VAV system cooling and TABS cooling act in parallel. Three past terms were used in Equations 4–6 to account for the building’s thermal lag.
The comparison between the TRNSYS and inverse model showed good agreement (Table 2) between the two models, with significantly shorter computational time for the inverse model, as reported earlier. The prediction errors in Table 2 were calculated according to Equation 10:

\[
\text{Relative RMSE} = \text{mean} \left( \frac{X_{\text{TRNSYS}} - X_{\text{inverse}}}{\max(X_{\text{TRNSYS}}) - \min(X_{\text{TRNSYS}})} \right)^2
\]

The importance of separating purely convective cooling rates from the internal loads (that have radiative and convective component) was tested by modifying Equations 4 – 6. When VAV system cooling rates and internal loads were combined into one term, the results showed relatively large prediction errors for both validation sets, as shown in Table 2. The return water temperature predictions were not strongly influenced by the modifications, since the zone loads are not a direct input for the water return temperature calculation (Equation 7).

The sensitivity analysis was performed on the second validation set (Figure 4b), demonstrating the influence that the number of past terms has on the prediction accuracy. The results showed that using three past terms offers a good balance between accuracy and computational speed for this particular room (Table 3).

<table>
<thead>
<tr>
<th>Number of past terms</th>
<th>Relative RMSE</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$T_z$</td>
</tr>
<tr>
<td>2</td>
<td>0.0308</td>
</tr>
<tr>
<td>3</td>
<td>0.0011</td>
</tr>
<tr>
<td>4</td>
<td>0.0012</td>
</tr>
<tr>
<td>6</td>
<td>0.0024</td>
</tr>
<tr>
<td>8</td>
<td>0.0007</td>
</tr>
</tbody>
</table>

2.1.4. Inverse model for humidity response

When humidity in the room needs to be controlled, the modeling environment uses the inverse model for humidity response. The model is also based on a transfer function,
with the zone’s humidity predicted as a function of the past humidity and past and current latent loads:

\[ w_z = \sum_{t=1}^{n} g^t w_z^t + \sum_{t=0}^{n} h^t L_{gain}^t \] (11)

Coefficients \( g \) and \( h \) are found by a linear regression to TRNSYS data.

Similarly as for VAV system cooling rates, the latent load depends on the zone’s humidity and vice versa. Therefore, the iteration loop similar to the one shown in Figure 2 has been added to the hygric inverse model.

2.1.5. Validation of inverse model for humidity response

To validate the inverse model for humidity response, first, the coefficients \( g \) and \( h \) were found by the linear regression to TRNSYS training data shown in Figure 5a. The training data assumed internal latent gains (e.g. a latent gain per person is approximately 0.07 kg/h) and latent loads due to ventilation/infiltration. Another set of TRNSYS data shown in Figure 5b was used as the validation data set. The inverse model showed satisfactory agreement with TRNSYS data for this type of analysis, with relative RMS error of 0.038.

2.2. Energy consumption

2.2.1. Cooling power

Heat pumps are components that consume the largest amount of cooling system energy, with the heat pump coefficient of performance (COP) representing the ratio between a cooling rate and consumed power. Although the COP is a function of a part-load ratio, evaporator air/water inlet conditions and condenser water/air inlet conditions, in the literature it is common to find HVAC system analyses that use a constant COP. This simplification might be acceptable, for example, when comparing the energy consumption of buildings with different facades or shading options. However, when using MPC to minimize the energy consumption, a constant COP would not give the optimal solution since it would not capture dependence of the performance on temperatures and part-load ratios. To characterize the heat pump performance, one could use manufacturer’s performance curves for a specific heat pump. However, manufacturer’s data are very limited, usually specified as an energy efficiency ratio (EER) evaluated at a single operating condition, or a seasonal energy efficiency ratio (SEER) evaluated for a single indoor temperature and some range of outside temperatures. For many advanced systems that can operate at wide range of conditions and loads, these data are not sufficient. Also, it is shown in Zakula et al. (30) that modest over-sizing of a variable-speed heat pump can be desirable, which suggests that even conventional systems could benefit from detailed heat pump performance data.

To characterize the performance of a specific heat pump, the modeling environment presented here uses the results of the heat pump static optimization. The result are obtained using a relatively detailed heat pump simulation model (31), and an optimization method presented in Zakula et al. (30). To reduce computational time, the static optimization data are approximated with polynomial curves using the linear regression. Figure 6a shows the results of the water-to-air heat pump static optimization (black) and fitted values (red) for a single water return temperature. A polynomial of the third
order was fitted to specific power (1/COP) curves since it was found that the specific power data are easier to fit than the COP data. The fitted polynomial is a function of the part-load ratio \((Q/Q_{max})\), outside temperature, \(T_x\) and water return temperature \(T_{w, return}\).

When the outside temperature is lower than the water supply temperature, the heat pump could run in the economizer mode, with the compressor turned off. However, in a case of smaller temperature differences across the evaporator and condenser (between air/water and refrigerant), the fan power will significantly increase. It is, therefore, incorrect to assume that the economizer mode will always give the lowest total power consumption. The advantage of the detailed heat pump model is its ability to optimize the economizer mode performance, and to enable the selection of the appropriate mode (normal versus economizer). Figure 6b shows the results of the static optimization (black) for the same water-to-air heat pump, now running in the economizer mode. The results were fitted by a polynomial of the fourth order (red), but different than the normal mode, the COP values were now easier to fit than the specific power data. Having the detailed performance maps for both the normal and economizer mode enables the MPC optimization function to decide which mode consumes less power in cases when the outside temperature drops below the water supply temperature.

2.2.2. Transport power

When comparing the energy consumption between different HVAC systems (for example water and all-air systems), the difference in the transport power can be significant. Moreover, it was reported in Krarti and Henze (20) that excluding the transport power from the optimization can influence the optimization results and cost savings. Therefore, the MPC objective function should account for the transport power, especially when analyzing HVAC systems with a large transport power relative to the cooling/heating power.

The fan/pump power \(P_{trans}\) is a function of a fan/pump characteristic, flow rates and total pressure losses throughout a building. The power at design conditions can be calculated using manufacturer’s data, or alternatively, using the limitations given in ASHRAE Standard 90.1 (4). Compared to supply fans, return fans have lower power at design conditions due to lower pressure drops in return ducts.

An additional correlation is needed to calculate the power at flow rates different than designed. As explained in detail in Englander and Norford (10), the power for centrifugal devices varies as a cube of the flow only in cases where the pressure is solely a function of a flow. For these cases, the power goes to zero as the flow rate goes to zero, and can be described with the simple power law. For example, an exhaust fan is controlled based on the airflow, and needs to overcome only losses in the exhaust duct. However, for devices that are pressure-regulated, and have a certain pressure to overcome, the fan power does not go to zero as the airflow goes to zero, but instead has a certain offset. Assuming a simple cube correlation and the zero power under no-flow conditions can significantly underestimate the power at low flow rates, as shown in Figure 7. An example is a HVAC supply fan that has setpoint static pressure to overcome. Englander and Norford (10) developed a correlation (Equation 12) for the dimensionless fan power \(P/P_{design}\), a function of the dimensionless pressure setpoint \(p_{set}/p_{design}\), and the dimensionless airflow rate \(V/V_{design}\). The correlation showed extremely good predictions when compared to experimental measurements for a VAV supply fan. While the modeling environment
uses Equation 12 to calculate the supply fan power, the return fans are modeled using the simple power law, where $Power = \text{Constant} \cdot (\text{Airflow})^3$. The simplified approach is appropriate for return fans since they are usually controlled based on the flow rather than the pressure setpoint, having a zero power for no-flow conditions. The same power law is used to model the pump power for off-design conditions.

\[
\frac{P}{P_{design}} = \left[ 1 - \left( \frac{1}{2} \frac{p_{set}}{p_{design}} \right)^{1.5} \right] \left[ \left( 1 - \frac{p_{set}}{p_{design}} \right) \left( \frac{V}{V_{design}} \right)^2 + \frac{p_{set}}{p_{design}} \right] \frac{V}{V_{design}} + \left( \frac{1}{2} \frac{p_{set}}{p_{design}} \right)^{1.5}
\]

(12)

2.3. Optimization of the objective function

An example of objective function optimization is shown here for a case of minimizing the total cooling energy consumption, and with the planning horizon of 24 hours. The objective function is a sum of the cooling power ($P_{hp}$), transport power required to deliver air/water to each zone ($P_{trans}$) and the temperature penalty related to thermal comfort ($T_{penalty}$). For each day the MATLAB function optimizes 24-hour-ahead cooling rates using the objective function:

\[
OF = \sum_{t=0}^{23} (P_{hp} + P_{trans} + T_{penalty})
\]

(13)

where

\[
P_{hp} = Q_{cc} \cdot COP \left( \frac{Q}{Q_{max}}, T_{fluid,c,in}, T_{fluid,c,in} \right)
\]

(14)

if $T_o < T_{llim}$

\[
T_{penalty} = F_{penalty} (T_{llim} - T_o)^2
\]

(15)

if $T_o > T_{ulim}$

\[
T_{penalty} = F_{penalty} (T_o - T_{ulim})^2
\]

(16)

The function $T_{penalty}$ penalizes values of the operative temperature that are outside the given limits. When choosing the appropriate penalty factor $F_{penalty}$, the amount of energy cost added to the penalty function (for the excursion just outside the comfort bounds) should be larger than the energy cost required to run the chiller to prevent that excursion (11). For example, assume that the chiller uses 200 W at the lowest compressor speed and under the most conditions. In that case, $F_{penalty} = 800$ W/K$^2$ means that if the operative temperature drifts 0.5 K outside the comfort region, the cost of running the compressor will be lower than letting the temperature drift any further.

2.4. Optimization algorithm

Three different configurations were considered for the optimization algorithm, shown here as an example of optimizing the cooling system with TABS for the lowest electricity consumption. The optimization parameters were chiller cooling rates ($Q_{TABS}$) in each hour of the following 24 hours.

Configuration A combines GenOpt (28) as the optimization engine with TRNSYS for the building’s thermal and hygric response (Figure 8). The objective function is calculated in MATLAB, but could also be calculated by developing a new module (type) in
TRNSYS. GenOpt is a general optimization software that can be linked to other computer programs (including MATLAB, TRNSYS and EnergyPlus). Detailed instructions for the connection with TRNSYS can be found in Kummert (21). The optimization method chosen from the GenOpt library is the combination of Particle Swarm Optimization (PSO) and Hooke-Jeeves Pattern Search (HJPS). This method, as well as some other non-gradient optimization methods, are recommended in the software manual (28) for optimization problems that calculate the objective function using building modeling programs. PSO, inspired by a social behavior of, for example, bird flocking or fish schooling, has a swarm of particles moving around the search space, where the movements of individual particles are influenced by the improvements discovered by others in the swarm. This method is used in the first optimization stage to find an appropriate starting point for the HJPS method. HJPS, similar to the MATLAB built-in pattern search, takes steps in different search directions, modifying the step size and search direction every time a lower objective function cannot be found using the current step size.

In configuration B, the TRNSYS building model is replaced with the inverse model. The optimization algorithm combines GenOpt and MATLAB, where MATLAB is evaluating the objective function and building response (Figure 9). The TRNSYS model is still used in the pre-optimization phase to create training data for the calculation of the inverse model coefficients.

Configuration C is using only MATLAB (Figure 10), which reduces the model complexity and computational time. The optimization method is MATLAB’s built-in pattern search algorithm, a non-gradient optimization method that does not guarantee finding the global minima (but neither do the gradient-based methods). However, as described in more detail in Wetter (28), gradient-based methods are not particularly suitable for the problems where the objective function is determined using building models, and can therefore be very dependent on building model tolerances. Furthermore, the chances of finding the global minima with the pattern-search method can be improved by choosing an appropriately large initial step and/or starting with a different initial point.

After analyzing all three configurations, configuration C was chosen for MPC analysis due to its simplicity and computational speed. Configuration A was abandoned due to its issues with initialization of variables and computational speed. This configuration was somewhat similar to the MPC models found in the literature, for which reported computational times were not suitable for fast simulation and implementation in real buildings. Moreover, connecting GenOpt, MATLAB and TRNSYS, or even GenOpt and TRNSYS would add to model complexity and possibly further degrade computational speed. Setting the GenOpt-MATLAB connection for configuration B was notably more challenging than setting the whole optimization in MATLAB (configuration C). Both configuration B and C predicted night precooling as the optimal strategy, with marginal differences in the energy consumption between the two configurations. However, while it took 30 s to optimize TABS cooling rates over 24 hours with configuration C, the computational time with configuration B (with default GenOpt optimization settings) was approximately 45 minutes.

In the post-optimization phase, after finding the optimal cooling rates, the second TRNSYS model simulates the behavior of a virtual building using the optimal values as inputs. Although slower than the inverse model, the TRNSYS model predicts building response with better accuracy. The accurate predictions of model outputs (mainly temperature responses) are important for the next optimization steps since the inverse
model requires the knowledge of the building's history.

The MPC schematic that optimizes VAV system cooling strategy would be very similar, with the VAV system cooling rate delivered to the room as the optimization parameter, and the airflow rates as inputs to the virtual building (the supply temperature for the VAV system is assumed to be constant).

3. Conclusion

The modeling environment (ME) presented here is developed to simulate buildings operated under MPC. The ME combines the optimization algorithm in MATLAB with two building models of different complexities, the building inverse model (based on transfer functions) and model from the commercially available building simulation program TRNSYS. Two additional modeling approaches are described, but were abandoned due to model complexity, issues with initialization of variables, and computational speed. The MATLAB optimization algorithm uses the pattern-search optimization method to find the optimal control strategy based on a predicted weather and internal gain forecast. The planning and execution horizon can be set depending on the accuracy of weather and load predictions. For example, in simulations were one has perfect predictions, both the planning and execution horizon are usually set to 24 hours. In a real building, the execution horizon would be shortened to account for unexpected changes in weather forecast, internal loads and building responses.

To predict the building's thermal and hygric response on the proposed strategy, the optimization algorithm uses the inverse model based on transfer functions. The temperature/humidity is calculated as the sum of previous temperatures/humidities, and current and previous loads (e.g. cooling/heating rates, internal gains, latent gains). The number of past terms increases with the heaviness of the building structure. The inverse model coefficients are calculated using training data from TRNSYS or from a real building. It was demonstrated that, compared to the TRNSYS model, the inverse model can give very good predictions in significantly shorter time. After the optimization, the optimal values are sent to the virtual building modeled in TRNSYS. Although slower than the inverse model, the TRNSYS model allows the prediction of the building's response to the optimal control strategy with better accuracy. Having accurate predictions in the post-optimization phase is important since the TRNSYS model outputs (thermal and hygric history) are used by the inverse model in the next optimization step. In a real building, the TRNSYS model would be replaced with the response from the building automation system (BAS).

The example of the ME setup was shown for the cooling system with thermally activated building surfaces (TABS) and variable-air-volume (VAV) system operated under MPC, and with the objective of optimizing the daily electricity consumption. The objective function was the sum of the heat pump power required for air/water cooling, transport power and temperature penalty, where the temperature penalty ensured that the controlled variable (operative temperature) is inside the desired comfort range. The heat pump performance was optimized using the heat pump model developed from first principles. The static optimization was decoupled from the building optimization, and the optimized data were used in the ME by fitting the polynomials to the heat pump optimization results. The heat pump performance was also optimized for the economizer.
mode when the outside temperature was lower than the water temperature. Altema-

tively, one could use manufacturer’s heat pump maps for a specific heat pump.

Although MPC is an emerging technology in the building industry, its implementa-

and analysis is greatly inhibited due to the lack of commercially or publicly available 

software tools. The ME presented here is open to the public, and can be provided upon 

request. It allows for the systematic study of primary factors influencing the dynamic 

control and savings potential for an individual building. Complex connections between 

building energy softwares and optimization algorithms are avoided by employing the 

simplified building model for the optimization. Except with TRNSYS, the ME can be 

used with other building softwares as well, as long as they account for building dynamics 

(e.g. EnergyPlus). Furthermore, the use of the simplified building model resulted in 

considerably shorter computational times then those found in the literature. For example, 

it took 30-60 seconds to optimize 24 cooling rates for a single-zone for a typical summer 

day (on the Windows 7 platform operated on Intel i5, 2.3GHz dual-core processor). 

Computational time is especially relevant for the implementation in real buildings, for 

which the optimization needs to be performed each hour (or shorter) due to weather 

and load uncertainties. Finally, the ME is highly modular, enabling simulation and 

optimization of different HVAC systems and optimization objectives. The use of the ME 

for MPC analyses was demonstrated in the accompanying paper (32) for a VAV system, 

VRF system and parallel TABS and DOAS system. However, it can be expanded to 

other systems, such as chilled beams, fan-coils, active thermal storage, ground source heat 

exchanges, etc. The ME can also be employed to determine the cost penalty associated 

with the use of near-optimal control, or in parallel systems to determine the right split 

between each system, e.g. between TABS and air-system heating/cooling. In the ongoing 

work, the ME is used to optimize the cooling system for a building that participates in 

the energy market by providing the ancillary services.

| Nomenclature |
|-----------------|----------------|---------------|
| **Symbol** | **Description** | **Unit** |
| COP | coefficient of performance | - |
| c | specific heat | J/(kgK) |
| L | latent load | kgw/s |
| m | mass flow rate | kg/s |
| OF | objective function | - |
| P | power | W |
| p | pressure | Pa |
| Q | heating or cooling rate | W |
| T | temperature | °C |
| V | volume flow rate | m³/s |
| w | absolute humidity | kgw/kgair |
| ρ | density | kg/m³ |
Subscripts

adj | adjacent room | max | maximal
---|---|---|---
air | air | o | operative
as | assumed | opt | optimal
avg | average | rad | radiative
c | condensing | return | return (water)
ce | cooling coil | s | supply (air)
conv | convective | supply | supply (water)
e | evaporating | trans | transport
hp | heat pump | ulim | upper limit
i | internal | w | water
in | inlet | x | ambient
llim | lower limit | z | zone
m | measured

References

Appendix - Experimental room description

Figure 1: Experimental room setup
Table 4: Experimental room construction

<table>
<thead>
<tr>
<th></th>
<th>Construction layers (inside to outside)</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>A. Wall</strong></td>
<td>0.016 m gypsum board</td>
</tr>
<tr>
<td></td>
<td>0.089 m air gap and 0.05 × 0.1 m stud wall</td>
</tr>
<tr>
<td></td>
<td>0.11 m polyisocyanurate foam R-30</td>
</tr>
<tr>
<td></td>
<td>0.016 m gypsum board</td>
</tr>
<tr>
<td><strong>B. Ceiling</strong></td>
<td>0.016 m gypsum board</td>
</tr>
<tr>
<td></td>
<td>0.140 m air gap and 0.05 × 0.15 m joists</td>
</tr>
<tr>
<td></td>
<td>0.11 m polyisocyanurate foam R-30</td>
</tr>
<tr>
<td></td>
<td>0.013 m plywood</td>
</tr>
<tr>
<td><strong>C. Window</strong></td>
<td>Three 1.12 × 1.17 m double pain windows separated by 0.09 m frames</td>
</tr>
<tr>
<td><strong>D. Floor</strong></td>
<td>Vinyl tile floor</td>
</tr>
<tr>
<td></td>
<td>0.025 m plywood</td>
</tr>
<tr>
<td></td>
<td>0.09 m floor joists with 0.076 m polyisocyanurate foam R-20</td>
</tr>
<tr>
<td></td>
<td>Existing concrete floor</td>
</tr>
<tr>
<td><strong>E. TABS</strong></td>
<td>Three layers of 0.044 m concrete pavers</td>
</tr>
<tr>
<td></td>
<td>0.00076 m aluminum</td>
</tr>
<tr>
<td></td>
<td>0.04 m plywood subfloor with 0.013 m PEX pipes (0.3 m center-to-center spacing)</td>
</tr>
<tr>
<td></td>
<td>D. Floor</td>
</tr>
</tbody>
</table>
Table 1: Relative RMS prediction errors for comparison between TRNSYS model and experimental measurements

<table>
<thead>
<tr>
<th></th>
<th>Relative RMSE</th>
</tr>
</thead>
<tbody>
<tr>
<td>$T_z$</td>
<td>0.0512</td>
</tr>
<tr>
<td>$T_{w,return}$</td>
<td>0.0040</td>
</tr>
<tr>
<td>$T_{floor}$</td>
<td>0.0339</td>
</tr>
<tr>
<td>$Q_{TABS}$</td>
<td>0.0256</td>
</tr>
</tbody>
</table>

Table 2: Relative RMS prediction errors for comparison between TRNSYS and inverse model (original and modified) on two validation data sets

<table>
<thead>
<tr>
<th>Validation set</th>
<th>Relative RMSE</th>
<th>$T_z$</th>
<th>$T_o$</th>
<th>$T_{floor}$</th>
<th>$T_{w,return}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1, original</td>
<td></td>
<td>0.0008</td>
<td>0.0006</td>
<td>0.0029</td>
<td></td>
</tr>
<tr>
<td>2, original</td>
<td></td>
<td>0.0011</td>
<td>0.0011</td>
<td>0.0049</td>
<td>0.015</td>
</tr>
<tr>
<td>1, modified</td>
<td></td>
<td>0.0169</td>
<td>0.0089</td>
<td>0.0026</td>
<td></td>
</tr>
<tr>
<td>2, modified</td>
<td></td>
<td>0.0322</td>
<td>0.0165</td>
<td>0.0062</td>
<td>0.015</td>
</tr>
</tbody>
</table>

Table 3: Sensitivity analysis for number of past terms in inverse model

<table>
<thead>
<tr>
<th>Number of past terms</th>
<th>Relative RMSE</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$T_z$</td>
</tr>
<tr>
<td>2</td>
<td>0.0308</td>
</tr>
<tr>
<td>3</td>
<td>0.0011</td>
</tr>
<tr>
<td>4</td>
<td>0.0012</td>
</tr>
<tr>
<td>6</td>
<td>0.0024</td>
</tr>
<tr>
<td>8</td>
<td>0.0007</td>
</tr>
</tbody>
</table>
### Table 4: Experimental room construction

<table>
<thead>
<tr>
<th>Construction layers (inside to outside)</th>
<th>A. Wall</th>
<th>B. Ceiling</th>
<th>C. Window</th>
<th>D. Floor</th>
<th>E. TABS</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0.016 m gypsum board</td>
<td>0.016 m gypsum board</td>
<td>Three 1.12 × 1.17 m double pain windows separated by 0.09 m frames</td>
<td>Vinyl tile floor</td>
<td>Three layers of 0.044 m concrete pavers</td>
</tr>
<tr>
<td></td>
<td>0.089 m air gap and 0.05 × 0.1 m stud wall</td>
<td>0.140 m air gap and 0.05 × 0.15 m joists</td>
<td></td>
<td>0.025 m plywood</td>
<td>0.00076 m aluminum</td>
</tr>
<tr>
<td></td>
<td>0.11 m polyisocyanurate foam R-30</td>
<td>0.11 m polyisocyanurate foam R-30</td>
<td></td>
<td>0.09 m floor joists with 0.076 m polyisocyanurate foam R-20</td>
<td>0.04 m plywood subfloor with 0.013 m PEX pipes (0.3 m center-to-center spacing)</td>
</tr>
<tr>
<td></td>
<td>0.016 m gypsum board</td>
<td></td>
<td></td>
<td>Existing concrete floor</td>
<td>D. Floor</td>
</tr>
</tbody>
</table>

**Notes:**
- Tabulated values are in meters (m) unless otherwise specified.
List of figures

Figure 1: Scheme of modeling environment for MPC simulation

Figure 2: Iteration loop for VAV system cooling rates

Figure 3: TABS cooling rate (blue), VAV system cooling/heating rate (green) and internal load (red) profiles for the training data set

Figure 4: TABS cooling rate (blue), VAV system cooling rate (green) and internal load (red) profiles for (a) validation set 1 and (b) validation set 2

Figure 5: Latent loads for (a) training data set and (b) validation data set

Figure 6: (a) Third-order polynomial fit (red) to optimized specific power (black) in compressor-on mode and (b) fourth-order polynomial fit (red) to optimized COP (black) in economizer mode. Results are shown for water-to-air heat pump and water return temperature of 17°C.

Figure 7: Dimensionless fan power versus dimensionless air flow relation for VAV and ventilation system supply fan, with (red) and without (black) static pressure setpoint

Figure 8: MPC algorithm setup A

Figure 9: MPC algorithm setup B

Figure 10: MPC algorithm setup C

Figure 11 (Appendix): Experimental room setup
Optimal control settings

Optimization algorithm

Assume cooling rates

Predict building response (operative temperature)

Calculate energy consumption (cooling + transport)

Minimize objective function (energy + comfort)

Real or virtual building

Building state
Assume $T_{z,as}$

$$Q_{\text{conv}} = 0.5Q_i + m_{\text{air}} c_{\text{air}} (T_s - T_{z,as})$$

$$T_z = \sum_{t=1}^{n} a_z t_{z}^t + \sum_{t=0}^{n} b_z t_{adj}^t + \sum_{t=0}^{n} c_z t_{x}^t + \sum_{t=0}^{n} d_z Q_{\text{TABS}}^t + \sum_{t=0}^{n} e_z Q_{\text{conv}}^t + \sum_{t=0}^{n} f_z Q_{\text{rad}}^t$$

Is $T_{z,as} = T_z$?

**YES**

**END**

**NO**

**START**
Figure(s)

- $Q_i$
- $Q_{TABS}$
- $Q_{air}$

$(W)$

Time (h)
Figure(s)
Figure(s)

- w/o static pressure set point
- w static pressure set point
Optimization algorithm

- GENOPT
  - Assume cooling rates
  - Minimize objective function

- MATLAB
  - Calculate objective function

- TRNSYS I
  - Predict building response

Building state:
- \([T_z,1-24, T_o,1-24, T_{floor,1-24}, T_{w,return,1-24}]\)

Objective function:
- \([T_{o,1} \cdots T_{o,24}]\)
- \([T_{w,return,1} \cdots T_{w,return,24}]\)

Optimal control settings:
- \([T_{w,supply,1} \cdots T_{w,supply,24}]\)
- \([m_{w,supply,1} \cdots m_{w,supply,24}]\)

Virtual building

Figure(s)
Optimal control settings

Optimization algorithm

GENOPT
Assume cooling rates
Minimize objective function

MATLAB
Predict building response
Calculate objective function

Building state

[Q_{TABS,1} \ldots Q_{TABS,24}]

TRNSYS II
Virtual building

Objective function

[T_z,1-24, T_o,1-24, T_{floor,1-24}, T_{w,return,1-24}]

[Q_{TABS,1} \ldots Q_{TABS,24}]

[Q_{TABS,1} \ldots Q_{TABS,24}]

Optimal control settings

[T_{w,\text{supply},1} \ldots T_{w,\text{supply},24}]

[m_{w,\text{supply},1} \ldots m_{w,\text{supply},24}]
Optimal control settings

TRNSYS II

Virtual building

MATLAB

Assume cooling rates
Predict building response
Calculate objective function
Minimize objective function

Optimization algorithm

Building state

\[ [T_{z,1-24}, T_{o,1-24}, T_{f,1-24}, T_{w,return,1-24}] \]
Figure(s)
A novel simulation tool for buildings that employ model predictive control is presented.

The tool enables a systematic study of primary factors influencing dynamic controls and the savings potential, and is available upon request.

The tool does not require source code modifications or complex connections between programs.

The use demonstrated on the advanced system with radiant cooling and optimization of energy consumption.