Digital Twin Design with On-Line Calibration for HVAC Systems in Buildings

Christian Vering¹, Sebastian Borges¹, Daniel Coakley², Hannah Krützfeldt¹, Philipp Mehrfeld¹, Dirk Müller¹,
¹RWTH Aachen University, Institute for Energy Efficient Buildings and Indoor Climate, Aachen, Germany
²Mitsubishi Electric R&D Centre Europe, Livingston, United Kingdom

Abstract

Digital twins of HVAC systems show great potential to increase efficiency through operational optimization and predictive maintenance of real processes. We introduce an IoT-based framework that allows the communication between a heat pump and its digital twin as well as further clients like a time series data base and supervisor dashboard. The obtained data is used to calibrate on-line the digital twin. Recalibration is performed to continuously improve the results. Using RMSE for evaluation and system supply temperature as target variable, we meet experimental data with RMSE<1 K automatically within a monthly time horizon.

Key Innovations

- Proposal of scalable application of digital twins
- Evaluation of heat pump recalibration in a period of one month
- Effect of plant aging on simulation models

Practical Implications

Our method is implemented in Python and can be applied to other scenarios using FMI in the Docker environment to couple further models or even dashboards. The optimization algorithm automatically calibrates the model according to measured data and is robust to different datasets. Thus, we provide an easy-to-deploy method to systematically roll out digital twins.

Introduction

As part of climate targets of the Paris Climate Agreement, the EU has pledged to reduce their emissions by 80 % to 95 % by 2050 compared to 1990 levels from which 30 % are related to buildings. Among other important adjustment levers, reaching this ambitious goal requires an almost climate-neutral building stock by 2050 and thus a sustainable energy supply for the whole building sector. To this end, particular efforts need to be made to exploit the saving potentials of building energy supply systems. (Bundesministerium, 2016)

Saving potentials can be exploited for example by updating conventional technologies by new systems. In this context, the heat pump is considered a key technology for a sustainable building heat supply (Bundesministerium, 2016). Furthermore, optimization of operation can offer saving potentials (Drgona, 2020). Operational optimization has proven to be promising for existing as well as new systems. One possibility to implement operation optimizations are digital twins (Vering, 2019). Thus, digital twins gain increasing interest in research and industry since recent years (Bauer, 2020). They enable the digital representation of real plants and processes and can map the characteristics as well as the operational behavior (Schleich, 2017). Digital twins have the potential to offer energy saving potentials for predictive maintenance (Vering, 2019).

The basis of digital twins is the use of models to describe and predict the behavior of real systems. Sufficient reliability and robustness of the models is therefore essential for the application of digital twins. For this purpose, a calibration of the simulation models is necessary. The goal of calibration is to minimize the difference between simulated and measured data by variations of certain model parameters (so-called tuner parameters) (Mehrfeld, 2021). The minimization can be solved automatically by implementing mathematical optimization methods. However, aging of machinery, e.g. due to physical object modifications, can change the system behavior. This leads to aging of simulation models, which were previously calibrated using measured data that does not fit to the current system behavior anymore. Thus, the prediction accuracy of existing models might reduce significantly (Chong, 2019).

According to Table 1, calibration methods can be distinguished in four groups: (1) manual calibration, (2) graphical calibration, (3) analytical calibration and (4) automated calibration. All methods aim for different accuracies, but also require a different execution effort. In general, all methods are able to offer high model accuracy, if the method suits the calibration problem. Automated calibration outperforms the other methods in terms of implementation time.

<table>
<thead>
<tr>
<th>Method</th>
<th>Accuracy</th>
<th>Implementation time</th>
</tr>
</thead>
<tbody>
<tr>
<td>Manual</td>
<td>+/-</td>
<td>-</td>
</tr>
<tr>
<td>Graphical</td>
<td>+/-</td>
<td>-</td>
</tr>
<tr>
<td>Analytical</td>
<td>+/-</td>
<td>-</td>
</tr>
<tr>
<td>Automated</td>
<td>+*</td>
<td>+</td>
</tr>
</tbody>
</table>

* Independent of technical experts

Refining the first three calibration methods, approaches of a repeated calibration (recalibration or on-line calibration) are already discussed in the literature, but mostly refer to
specific, technical applications (Tügel, 2012; Chong, 2019; Doekemeijer, 2018). A modular use of simulation models for air-source heat pumps is not addressed, so far. Therefore, the overall goal of this contribution is to develop a modular framework for automated recalibration of digital twins. Through a generic structure and object-oriented programming, arbitrary simulation models can be integrated into the framework, thus enabling a wide variety of applications in practice. Furthermore, a modular integration of databases is set up.

To show the applicability of the framework for the recalibration of digital twins, a use case consisting of a building and an air-source heat pump is investigated for a period of one month. For this purpose, the heat pump model is created and parameterized with the help of the Modelica model library called AixLib (Müller, 2016). We test the functionality of the framework and analyze interdependencies of calibration classes and influences of recalibrating the simulation models on the results. This publication is organized as follows:

- Chapter 2 shows the framework of recalibration and explains all steps towards fully automated on-line calibration methods.
- Chapter 3 describes the heat pump use case, the simulation model parameters, the tuning parameters and the conducted sensitivity analysis.
- Chapter 4 presents calibration results.
- Chapter 5 discusses the application of the framework, on the basis of which application possibilities and limitations are shown.
- Chapter 6 draws conclusions and introduces further work.

On-line Calibration

In this paper, we present a framework for repeated calibration (on-line calibration or recalibration) of simulation models (ReCaMo). The framework is applied to an air-source heat pump providing heat for a real building to show the functionality of recalibration. We use measurement data of three days to recalibrate the heat pump model. This is done on-line ensuring to predict the real behaviour of the system uninterrupted by external influences that occur during operation on the heat pump. Thus, ReCaMo can, for example, increase model prediction accuracy (1) or even mitigate the effect of aging of simulation models (2). Within the short field test period of one month, we are able to prove (1) and give a promising outlook that we are one step closer to (2).

The structure of the framework is generic to allow the application of any type of simulation model by using FMI and thus to ensure modularity. ReCaMo is implemented in Python and its structure is schematically depicted in Figure 1. Hereafter, the six steps are explained separately in detail.

Figure 1: Structure of the ReCaMo-framework for the recalibration of digital twins.

Step 1: Instantiation

At First, the ReCaMo framework is instantiated by a configuration file. This file contains information about the simulation model (e.g. model type), the configuration for simulation, sensitivity analysis and calibration as well as the data source being used. The configuration file is read and the settings are applied in the framework. In this paper, we show its application on an air-source heat pump. In general, the framework should be applicable to simulations models that fit pyFMI standard (Storek, 2019).

To ensure modular use of simulation models, an API (application programming interface) class is provided for the link between the framework and the selected model type. The methods of the program classes of different model types can thus be individualized.

Step 2: Data Acquisition

After instantiation, measurement data for in- and outputs of the model is extracted (cf. Figure 2) and cleaned according to the specifications in the configuration file (e.g. start and end time, designations of database and measurement point). Furthermore, a data mapping is provided to support the user by mapping of the measuring point designations into the database and the corresponding data points in the simulation model. The mapping is defined once for each simulation model by the user in the configuration file. Figure 2 indicates how measurement data could look like in the data acquisition step. It shows the measured supply temperature of a building for an exemplary day. Obviously the data is highly dynamic and the courses over a day differ from each other. This dataset will be applied to ReCaMo.

Within ReCaMo an InfluxDB database is used to store and handle data. Due to the modular structure, it is also possible to easily integrate further databases in ReCaMo such as SQL. InfluxDB is an open-source time series database, which was designed to handle time series data efficiently. The SQL-like query language is easy to learn and understand. (Naqvi, 2017)
Step 3: Simulation and Preparation
Following the data acquisition, a first simulation of the model is performed. ReCaMo uses all initial parameter values specified in the model. The results of the simulation are subsequently used in the following preparation step. During this preparation step, the so-called goals, tuner parameters and calibration classes are defined.

The proposed concept of goals represent variables for the comparison of measured and simulated values in a tabular form (e.g. of a supply temperature in a heating circuit).

All tuner parameters specified in the simulation model, including the initial and allowed minimal and maximal values, are automatically identified and read by the framework. Within the calibration process, tuner parameters are varied to minimize a predefined evaluation metric in a certain time period. Since each period might have unique characteristics, calibration classes can be introduced in ReCaMo.

The proposed concept of Calibration classes (CCs) describes different operation modes that are calibrated individually using the corresponding time intervals. Operation modes for different CCs could e.g. be on-/off-modes and the differentiation between transient and steady state operation. The CCs thus ensure a piecewise calibration of the entire examination area (e.g. the calibration of one day). Each CC contains its own related goals and tuner parameters. Depending on the application, several CCs with different tuner parameters and target values can be defined, as the sensitivity of the tuner parameters may differ for different operation modes.

Step 4: Sensitivity Analysis
The sensitivity analysis gives information about the extent to which a change of specific parameters (tuner parameters) influences the simulation results. Thus, in step 4 ReCaMo identifies the impact of changes in parameters on the simulation model output. Parameters without significant impact are not considered during calibration, thus simplifying the process. To analyse the sensitivity in simulation models before calibration and find the relevant parameters in ReCaMo, we apply the Morris method (Morris, 1991). This method represents a one-factor-at-a-time method. The input variables remain unchanged for each parameter variation.

According to the analysis, the sensitive parameters for calibration are selected semi-automatically afterwards. For this purpose, the user has to identify all relevant tuner parameters in the simulation model and distinguish between sensitive and non-sensitive parameters. The sensitive parameters then have to be declared as tuner parameters. This first step is done manually and only once for each simulation model. From this point on, the Morris method can be used to automatically select the sensitive tuner parameters according to the settings of the configuration file. If automation is not desired by the user, the tuner parameters can also be set manually in the configuration file (in the case the majority of the sensitive parameters are already known). The sensitivity analysis step can thus be used optionally and can be deactivated depending on the scope and application.

Step 5: Calibration
The aim of calibration is to increase the accuracy of a model by adjusting the tuner parameters through parameter variation, and thereby reduce the deviation between simulation $\hat{x}_i$ and measured data $\tilde{x}_i$. In the case of automated calibration methods, calibration can be understood as minimization of a chosen objective function (deviation between measured and simulated data) and thus can be formulated as a mathematical optimization problem. Common key performance indicators (so called “metrics”) to evaluate the deviation are shown in Table 2. The “Mean Absolute Error” (MAE) and “Root Mean Square Error” (RMSE) have the unit of the target variable (e.g. K for supply temperature). By normalizing the error measures, the dimensionless “Normalized Root Mean Square Error” (NRMSE) and “Coefficient of variation of Root Mean Square Error” (CV(RMSE)) are obtained. Therefore, the mean value $\tilde{x}$ for the CV(RMSE) and the range $\tilde{x}\text{max} - \tilde{x}\text{min}$ for the NRMSE are used as reference values. All above mentioned methods are implemented in our optimization framework. Within this paper, we use RMSE and CV(RMSE), to evaluate the calibration results.

The modular structure of ReCaMo also allows the use of different optimization methods. In this context, we apply Differential Evolution (DE), which represents a method of the evolutionary algorithms (Storn, 1997).

In order to limit a scatter of the optimized values of the tuner parameters and obtain more consistent results, a deviation penalty function is implemented in the scope of this work:

$$ e = e_{\text{opt}} + \sum_{k=1}^{n_e} w_k (b_k - b_k^*)^2 \quad (1) $$

The objective function $e_{\text{opt}}$ (e.g. the RMSE) is extended by a penalty factor. This penalty factor is calculated from the squared deviation of a tuner parameter $b_k$ to a benchmark $b_k^*$ and thus affects the value of the objective function. A benchmark represents the solution of all tuner parameters of a recalibration, which has the lowest value.
of the metric of all recalibrations so far. This is updated as soon as the value of the metric of a recalibration is lower compared to the current benchmark. \( n \) represents the number of all tuner parameters used for calibration and \( w_q \) denotes a constant weighting factor for each tuner parameter, which can be specified by the user in the configuration file.

Table 2: Metrics to evaluate the deviation between simulated \( x_i \) and measured \( \hat{x}_i \) results. \( n \) represents the number of sampling points.

<table>
<thead>
<tr>
<th>Metric</th>
<th>Formula</th>
<th>Shortcut</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mean absolute error</td>
<td>( \frac{\sum_{i=1}^{n}</td>
<td>x_i - \hat{x}_i</td>
</tr>
<tr>
<td>Root-mean-square error</td>
<td>( \sqrt{\frac{\sum_{i=1}^{n} (x_i - \hat{x}_i)^2}{n}} )</td>
<td>RMSE</td>
</tr>
<tr>
<td>Normalized root-mean-square error</td>
<td>( \frac{RMSE}{\hat{x}<em>{\text{max}} - \hat{x}</em>{\text{min}}} )</td>
<td>NRMSE</td>
</tr>
<tr>
<td>Coefficient of variation of root-mean-square error</td>
<td>( \frac{RMSE}{\hat{x}} )</td>
<td>CV(RMSE)</td>
</tr>
</tbody>
</table>

The tuner parameters as well as the benchmark are normalized between 0 and 1 using their limit values from the model to ensure scale independence and thus make it possible to compare different tuner parameter sets with each other. According to this formulation, optimization runs are conducted automatically.

Step 6: Overwriting

The optimized tuner parameters, metrics and goals of each iteration are stored in the InfluxDB. The results of each recalibration are compared continuously. If the value of the metric decreases, all related results are overwritten in a separate file, so that the best performant parameters are not lost.

In the case that several CCs are used within one recalibration step (e.g. one day), it may occur that some tuner parameters are used in more than one CC. This can lead to different optimal values for these tuner parameters in the respective CCs. As a result, there are multiple solutions for tuner parameters within a recalibration step. Figure 3 shows the objective functions for two calibration classes using the same tuner parameters.

Apparently, the optimal values \( r_{\text{opt,1}} \) and \( r_{\text{opt,2}} \) of the tuner parameters differ after calibration for both classes. To determine the overall solution of a recalibration step, the results of the tuner parameters are averaged \( \bar{F} \) and the objective function is evaluated again. Due to averaging, the calculated solution is not the optimal solution for any of the CCs, unless the results of the CCs are the same. For further investigations, an additional optimization step might improve the averaging step to ensure global optimal tuner parameters with respect to all CCs. Averaging can be specifically prevented by not allowing overlapping of the sensitive tuner parameters during configuration. In this case, the tuner parameters would have to be selected manually for each CC. Further work could deal with the automation of this functionality.

The last step of ReCaMo is to overwrite the latest tuner parameter, the entire process is repeated, and current measurement data is extracted again. The optimal time until next repetition is not investigated in the scope of this contribution. We recalibrate a heat pump model three times, in each week once, within one month according to field test data. This case study is introduced in the following Chapter.

Case Study: Heat Pump Calibration

The application of ReCaMo focusses on a heat pump, which is being modelled and recalibrated using the framework. We investigate one month of measurement data. A recalibration step covers one day from 00:00:00 h to 23:59:50 h with a sampling rate of 10 seconds. Within this contribution, we calibrate the heat pump model once based on the first day and we recalibrate the model according to two other days to show the potential of our method.

Figure 4 shows the heat pump that is modelled in Modelica using the AixLib (Müller, 2016) heat pump model. The refrigeration circuit is implemented as a black-box using a characteristic curve from the manufacturer's specifications and the heat exchangers as semi-empirical grey-box models. This type of model represents a compromise between computational effort and model accuracy.

Heat capacity and heat losses are considered in the heat exchangers. Heat source and heat sink of the heat pump are modelled using ideal mass flows. Pressure losses in the heat exchangers are not considered, yet, just as the heating water circuit and hence the water pump are not investigated, respectively. Icing effects and the control of the system including safety functions are also disregarded.
Figure 4: Scheme of the heat pump model, which is (re)calibrated using ReCaMo.

Figure 4 shows a scheme of the heat pump model with the basic components compressor, condenser, expansion valve and evaporator. For the simulation of the heat pump, the return temperature $T_{w1}$ (water inlet to the condenser), the ambient temperature $T_{a1}$ (air inlet to the evaporator), the water mass flow $\dot{m}_{w1}$, air mass flow $\dot{m}_{a1}$, and the relative speed of the compressor $u_{set}$ are required (circled dashed). The output variables of the model (circled solid) are the supply temperature $T_{w2}$ (water outlet from the condenser) and the electrical power $P_{el}$ of the heat pump compressor. Only the supply temperature is used for calibration, since the electrical power of the compressor is not measured separately. Using this setup and measurement data such as shown in Figure 2, we can use ReCaMo for calibration of the model.

Calibration Results

In the following, we evaluate the results from the ReCaMo framework for recalibration of digital twins by presenting a sensitivity analysis (I), the first calibration (II) and the recalibration (III) based on one month. In most cases, we use one CC for calibration purposes and a weighting factor for the deviation penalty function of $w = 0.2$ is set heuristically a priori. An adjustment of this weighting can be necessary, when ReCaMo is to be applied to further simulation model calibrations based on field test data.

Results I: Sensitivity analysis

The relevant tuner parameters from the model for the sensitivity analysis are shown in Table 3. They are evaluated within the Morris method using the absolute mean value of the elementary effects $\mu^*$, which represents the ratio of the variation of the output with respect to the variation of the input between two points within the examination range by applying a standardized relative amount (Morris, 1991). The number of parameter variations (called trajectories) is 100, the number of possible parameter values (called grid levels) is 4. The CV(RMSE) is chosen for the evaluation of the sensitivity of the supply temperature.

<table>
<thead>
<tr>
<th>Table 3: Notation of relevant tuner parameters.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Parameter</strong></td>
</tr>
<tr>
<td>$\omega$</td>
</tr>
<tr>
<td>$\Delta p_e$</td>
</tr>
<tr>
<td>$V_c$</td>
</tr>
<tr>
<td>$G_{e,a}$</td>
</tr>
<tr>
<td>$G_{e,j}$</td>
</tr>
<tr>
<td>$C_e$</td>
</tr>
</tbody>
</table>

Figure 5 shows the results of the absolute mean values $\mu^*$ of the sensitivity analysis with measurement data from the first day for the given tuner parameters. Evidently, the volume of the condenser $V_c$, the cut-off frequency $\omega$ and the heat loss parameters $G_{e,a}$ and $G_{e,j}$ dominate the impact on the target variable. The sensitivity of the heat capacity of the condenser $C_e$ is negligible. The absolute mean values of the evaporator-specific parameters ($V_c$, $G_{e,a}$, $G_{e,i}$, $C_e$) are barely sensitive to the target variable. This corresponds to the expectations, as the outlet temperature is directly connected only to the condenser and not the evaporator. Furthermore, the pressure losses in evaporator $\Delta p_e$ and condenser $\Delta p_c$ have no influence on the supply temperature. This is due to the fact, that the heating circuit and the fan are represented by ideal mass flow sources.

The results obtained on the other days of the study period match those obtained on day one (cf. Figure 5). According to the sensitivity analysis, further evaluations within the calibrations are based on the four highest ranked tuner parameters $\omega$, $V_c$, $G_{e,a}$ and $G_{e,j}$.

Results II: First calibration day

Figure 6 shows the measured and simulated data after the first calibration of the supply temperature of the first day of the investigation from 01:08:30 h to 06:03:30 h. The results of the simulation after calibration are similar to the measured data which yields an RMSE of the whole calibration day of only 0.60 K. Temperature peaks can be predicted by the model. Apparently, the temperatures are
rather underestimated in the simulation. This can be caused by the fact that a rotational speed dependency of the heat pump characteristic curve is only approximated by using \( u_{\text{net}} \), since only a characteristic curve at a fixed rotational speed is available for modelling the heat pump. This heat pump model can now be used as digital twin until the next calibration process is conducted. In this paper, we conduct a second calibration one week later.

Figure 6: Measured (blue) and simulated (red) supply temperature in Kelvin (goals) from 01:08:30 h until 06:03:30 h from day 1. RMSE of the whole calibration day: 0.60 K.

Results III: One month results

Figure 7 shows a decrease of the RMSE during the investigated period. Accordingly, a reduction of the metric on the following days of consideration can be achieved by recalibration. Thus, the accuracy of the simulation results increases continuously. This can be explained by the different measurement data and its information content for the ReCuMo procedure of the calibration days.

Table 4 shows the general settings of the framework and the results of the tuner parameters of the second day investigation.

Table 4: Recalibration with measurement data from recalibration day 7 using the RMSE.

<table>
<thead>
<tr>
<th>Setups</th>
<th>Results</th>
</tr>
</thead>
<tbody>
<tr>
<td>Target</td>
<td>( T_{w2} ) [K]</td>
</tr>
<tr>
<td>Start</td>
<td>00:03:20 h</td>
</tr>
<tr>
<td>End</td>
<td>23:59:50 h</td>
</tr>
<tr>
<td>Metric</td>
<td>RMSE</td>
</tr>
<tr>
<td></td>
<td>RMSE</td>
</tr>
</tbody>
</table>

Results IV: Use of penalty function

In the following, the influence of the weighting factor \( w \) of the squared deviation on the results of the tuner parameters is investigated. The recalibration with the target variable \( T_{w2} \) is carried out over all 31 days from 00:00 h to 24:00 h. Figure 8 shows the results of the tuner parameters with a weighting factor \( w = 0.0 \) (left) and \( w = 0.4 \) (right) and the corresponding boundaries of the tuner parameters (red line). When comparing the parameter values on the left and on the right, it can be observed that the scatter of the results is reduced by the deviation penalty function. The deviation penalty function can thus be used efficiently to contain the range of results of the tuner parameters.

However, a too high weighting factor will also reduce or even prevent necessary changes in the parameters during recalibration. Thus, with higher weighting factors the impact of the very first calibration increases, as the results of following recalibrations will strongly depend on it. If this first result is of poor quality, e.g. due to faults in the measurement data, this has a negative effect on all further calibration results. Therefore, the choice of the weighting factor \( w \) must be carefully chosen for each particular case.

Table 5 shows the results of the recalibration with a weighting factor \( w = 0.0 \) (left) and thus without application of the deviation penalty function as well as \( w = 0.4 \) (right) day 14. The results of the tuner parameters differ significantly, while the result for the CV(RMSE) is similar in both cases. The application of the deviation penalty function therefore does not necessarily have a negative effect on the calibration results.
Table 5: Results of recalibration day 14 without (w = 0.0) and with (w = 0.4) the deviation penalty function.

<table>
<thead>
<tr>
<th></th>
<th>Results day 14, w=0.0</th>
<th>Results day 14, w=0.4</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\omega$</td>
<td>0.10374</td>
<td>0.00688</td>
</tr>
<tr>
<td>$V_c$</td>
<td>0.00368</td>
<td>0.00048</td>
</tr>
<tr>
<td>$G_{c,a}$</td>
<td>0.55</td>
<td>57.06</td>
</tr>
<tr>
<td>$G_{c,i}$</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>CV(RMSE)</td>
<td>0.002469</td>
<td>0.002323</td>
</tr>
</tbody>
</table>

Results V: Use of calibration classes

Previous calibration results refer to calculations with one CC. The calibration is performed continuously using one day for training. Since the model behaviour and thus the results of the calibration depend on the operating mode of the system, it makes sense to examine the modes separately with respect to the calibration in different CCs.

The selection of the CCs is done based on the relative speed of the compressor. For the investigation, we chose the time interval between 01:08:30 h and 06:03:30 h of day 1. Figure 9 shows the measured supply temperature and the relative speed of the compressor in the investigated time interval at red highlighted background.

Figure 9: Measured data of supply temperature (top) and relative compressor speed (bottom) of day 1.

The classification of the CCs is based on the periods in which the compressor is switched on and off. For the investigated time interval, this gives 18 sections. By merging CCs of the same type, the number can be reduced to 2: compressor switched on and switched off.

The calibration of the investigated period shown in Figure 6 is used as the reference case with only one CC. The RMSE in this case is 0.73 K. By using the merged CCs (cf. Figure 10), an improvement of about 15 % of the metric can be achieved having an RMSE of 0.63 K. In addition, the results of the tuner parameters of the classes “compressor on” and “compressor off” are averaged (see section 5.6) to compare the RMSE with the calibration of a single CC.

Despite the decrease of the metric due to averaging (cf. Figure 3), the accuracy of the simulation results can be increased by using well-chosen CCs.

Figure 10: Goals of the investigated period with two calibration classes, compressor on (top) and compressor off (bottom). The grey areas are not taken into account, the red areas are the CCs, respectively. The right plot is a zoom of the highlighted area on the left.

Capabilities and Limitations

This section discusses the obtained results to show potential for refinement of the method.

Capabilities

The analysis shows that the calibration using the ReCaMo framework provides physically reasonable results. Furthermore, continuous recalibration of the simulation model within specific calibration intervals can continuously increase the performance. The use of CCs for different operating phases can further increase the simulation accuracy.

Due to the aging effects of machinery, updates of simulation models during operation is necessary. ReCaMo automatically recalibrates simulation models based on current measurement data after the configuration and instantiation. A sufficient quality of the simulation results can thus be ensured over the entire application period of the framework and the aging effects can be considered in the simulation model.

In the literature, the implementation of recalibration is limited to specific simulation models and applications. A continuous recalibration using the framework of the present study is designed for a modular use of simulation models. This ensures a generic usage of the framework. ReCaMo has already been successfully tested on several heat pumps internally. In order to ensure a user-friendly application, the stability of the framework is increased by the implementation of an error handling of the user inputs.

A deviation penalty function is implemented to limit scattering of the optimized tuner parameters obtained from calibration. The results are thus more physically reasonable.

Limitations

A potential use of the CCs is limited at this point to a manual identification of the operating phases of the measured data. For an automated usage of the CCs, the integration of an automated pattern recognition of the measurement data is necessary or a digital twin of the controller must be known a-priori.

So far, ReCaMo does not allow any intervention in the operating process of the real plant, e.g. by control signals or anything similar. The results of the calibrations can
provide information for control applications such as model predictive control or methods for forecasting and fault detection and diagnostics (FDD) of technical processes. Furthermore, the calibration is performed in fixed time intervals specified by the user. A deviation-based calibration, e.g. by real-time simulations and continuous control of a selected error measure between simulated and measured data, is not implemented, yet.

Equation 1 shows that the penalty factor is calculated for each tuner parameter of the calibration. Thus, the value of the penalty factor depends on the number of calibrated tuner parameters. In this contribution, four tuner parameters are calibrated. If an application of the ReCaMo aims e.g. at the simulation of building physics for building energy management, the number of sensitive tuner parameters can be significantly higher. This leads to an increase of the penalty factor, which leads to an increased limitation of the results of the tuner parameters by the deviation penalty function. In addition, the sensitivity of the tuner parameters is not considered when calculating the penalty factor. Each parameter is equally weighted within the deviation penalty function by the weighting factor \( w \). For a further usage of the deviation penalty function, the weighting factor \( w \) can be chosen depending on the number and sensitivity of the tuner parameters, which would improve the calibration process.

**Conclusion and Outlook**

Within this contribution, we introduce ReCaMo as a framework for repeated calibration of simulation models and apply it to a heat pump model. Using one month of field test data, we proved functionality of our framework and assessed our results using RMSE.

Comparing three different days, we calibrated a heat pump model from the AixLib utilizing an a-priori sensitivity analysis to systematically reduce calibration effort. Thus, we are able to decrease the RMSE of the heat pump supply temperature between the model and the simulation results from day to day, which was always below 1 K within our use case. To further improve our results we introduced calibration classes, which again decreased RMSE by considering whether the heat pump is switched on or switched off.

For future investigations, we suggest to enhance the choice of calibration classes to improve calibration results. Furthermore, ReCaMo should be tested with other system configurations to examine the modularity of the whole framework. Increasing the development of such frameworks and proving them in real world applications accelerates the introduction of digital twins in the building sector. This is important to significantly decrease emissions in that sector.

**Acknowledgement**

We gratefully acknowledge the support of Mitsubishi Electric R&D Centre Europe.

**References**


