Applying Machine Learning to Automate Calibration for Model Predictive Control of Building Energy Systems

Thomas Storek, Asad Esmailzadeh, Philipp Mehrfeld, Markus Schumacher,
Marc Baranski, Dirk Müller
RWTH Aachen University, E.ON Energy Research Center,
Institute for Energy Efficient Buildings and Indoor Climate, Aachen, Germany

Abstract

About 74 % of model calibrations happen manually. This work presents an automated calibration method. A key aspect of calibration is the identification of dominant model parameters, which for energy conversion systems, e.g. heat pumps, strongly depend on the operating state. Starting from energy monitoring data, we analyze the time series and identify characteristic operating periods. The latter can be a start-up phase, continuous operation or a cool down period etc. Training a decision tree classifier with manually assigned data, we process the entire monitoring data automatically and split the data into period specific subsets. Using the Morris-Method for sensitivity analysis enables a ranking of calibration parameters for each subset. Followed by successive calibrations where each only considers the most dominant model parameters, we tune the model. A cross validation finalizes the process.

Introduction

Today, the fraction used for heating, ventilation, air conditioning and refrigeration (HVACR) accounts about two-thirds of the building energy consumption. Hence, there is an urgent need for energy saving strategies, which is also a focus of energy policies in many countries. (Afram and Janabi-Sharifi (2014); Baldi et al. (2015); Killian and Kozek (2018))

Although "passive" approaches such as retrofitting may reduce the energy consumption for room heating, they are not able to guarantee acceptable comfort levels and tend to be cost intensive. In fact, all the advantages of improved building design may be lost or can become even contra productive, with significant reduction of indoor climate comfort by poor or simplified system control strategies. (Michailidis et al. (2018))

Optimizing the operation of building energy systems shows high potential for energy savings in the building sector (Fütterer et al. (2017)). However, due to the integration of renewable energy sources into building energy systems, the system complexity increases significantly. Hence, the reliable implementa-

tion of classical control approaches such as On/Off or proportional-integral-derivative (PID) control is cumbersome and can lead to low energy efficiency. Therefore, there is a need for advanced control approaches that are able to handle this complexity. One of the most promising approaches is model predictive control (MPC). In literature, energy savings of around 20 % to 35 % and reduced operating cost of up to 73 % are reported, when compared to classical approaches. Although MPC is a standard technique in plant automation after it was first introduced in the 1970s (Dittmar and Pfeiffer (2006)) and extensive research in the field of building energy management systems (BEMS) has been carried out, the practical application of MPC is quite rare and still objective of ongoing research. (Afram and Janabi-Sharifi (2014); Baldi et al. (2015); Killian and Kozek (2018))

In literature, modelling and calibration are considered as most important and, simultaneously, most time consuming parts. Nevertheless, despite the influence of non-linearities, disturbances, time-varying system dynamics and interdependencies in building energy systems (BES), according to Coakley et al. (2014) about 74 % of the calibrations in simulations happen manually using trial and error techniques. Besides the significant amount of time required, manual methods are highly dependent on the engineer's expertise and experience with the respective simulation model. This indicates the need for a simplified and more effective calibration procedure for dynamic models of energy conversion systems in order to pave the way for commercial MPC implementations.

Due to the high system complexity and the amount of model parameters involved, the calibration of BES-models represents an underdetermined optimization problem, which has a non-unique solutions. Reddy (2006), Coakley et al. (2014) and Fabrizio and Monetti (2015) provide valuable reviews about automating the calibration process for BES. Furthermore, Coakley et al. (2014) define seven issues concerning calibration of BES-models: Standardization, expenses, simplification, low quality of input data, consideration of uncertainty, identification of causes for model discrepancies and lack of automated calibra-

tion methods. The Bayesian calibration, which is gaining attention in the recent years, considers the uncertainty of model input data for calibration. Li et al. (2015) present a generic Bayesian approach for calibrating BES-models including a sensitivity analysis and a meta-model-based optimization of a linear regression model.

Even though the potential is not fully leveraged yet, applying optimization algorithms belongs to the state of the art in calibration technology, where the general idea is the minimization of the deviation between model output and corresponding measurements using gradient based algorithms. In practice these reach their limits for BES-models because the objective function converges either to local minima or does not converge at all. Therefore, we propose a simplification of the optimization problem by reducing the number of parameters that are calibrated simultaneously and split the overall process into multiple successive calibrations. In order to identify the most dominant parameters, we use a sensitivity analysis referring to Eisenhower et al. (2012). Supported by machine learning techniques we automate the process in order to reduce the effort for implementation and maintenance of MPC in BEMS.

Methodology

Although automated methods for model calibration bear major advantages compared to manual calibration, one major challenge is yet to be faced. In most cases the simultaneous tuning of multiple parameters in a complex BES-model will not lead to an optimal solution. This is mainly caused by the nature of non-linear optimization problems in energy systems which are often circumvented by using heuristic approaches and specialized solvers in order to provide satisfactory results. However, they are limited in its universal applicability, to the best of our knowledge, none of them can guarantee to find a global minimum for non-linear and non-convex calibration problems. Therefore, we present a new method for automated calibration that devides the calibration process into three sequential parts as illustrated schematically in figure 1. First, a clustering analysis and a classification of time series data is used to separate physical effects based on pre-selected features X_i . Second, a sensitivity analysis is performed to identify the most dominant parameters within the characteristic sequences. Finally a mathematical optimization is executed to adjust the parameters for each sequence so that the model discrepancy Δy can be minimized.

Figure 2 illustrates a typical supply and return temperature as well as the volume flow as time series data for a heat pump. One can observe that the temperature curves behave in certain patterns (in the following called effects) which occur repetitive. These effects are mainly caused by physical processes in BES. For instance, a valve opens or a pump starts operat-

Figure 1: Calibration method consisting of a classification of physical states, a sensitivity analysis and an optimization (Esmailzadeh (2018)).

Figure 2: Typical temperature curves (inlet and outlet) as well as the corresponding volume flow curve for a heat pump.

ing in a certain time step leading to a characteristic behavior of the observed temperature curves.

Simulation models usually reproduce these phenomena derived from geometrical and physical equations using different physical and non physical parameters for adjustments, which are only partly known from manufacturer's instructions or estimated based on the modeler's expertise. Nevertheless, by distinguishing between different classes with dominant physical effects we can assign selected parameters to classes and tune them separately. The advantage of this procedure is that a reduced number of parameters can be declared as variables during the optimization stage which improves the chance for convergence. To transfer this idea into an automated procedure, we use a combination of unsupervised and supervised learning algorithms, which will be explained in the following.

The category of unsupervised learning defines methods that can be applied to data sets in order to create clusters without being trained with labeled data previously. In contrast, supervised learning algorithms

require a training phase, in which related patterns between characteristics of the data and the a priori assigned classes are studied by an algorithm. These two types and their applications are often misunderstood in literature. Since both learning categories define fundamentally different procedures, they bear also different advantages and fields of application.

We use unsupervised cluster analysis to gain information about the given data set without involving human experience. The clustered data reveals not necessarily how many, but rather which kind of different physical effects we have to consider for building and calibrating models. The automatically found classes can be passed subsequently to a supervised classification procedure, which then trains an algorithm to identify the classes in a wider range of data. Thus, the unsupervised learning algorithm increases the degree of automation while the supervised learning method adds the ability for up-scaling. (Esmailzadeh (2018))

One of the most used clustering methods is the *K-means* algorithm, which creates groups of data points based on the euclidean distance between the centroid of the clusters and the considered data point in a multi-dimensional feature space (Béjar (2013)). The predefined features e.g. supply and return temperatures, pressures etc. are evaluated in every time step and the data points are assigned to the clusters accordingly. Since the coordinates of the cluster's centroid is shifted with each assignment, the assignments are iterated until the coordinates do not change anymore. The number of clusters is set a priori. As an extension of the sequential method we integrate an automated class creation process by using a *Mini Batch K-means* clustering algorithm. Béjar (2013) gives a detailed comparison between *K-means* and *Mini Batch K-means*. The *Mini Batch K-means* is a more efficient extension of the simple *K-means* algorithm, which evaluates a randomly selected excerpt of data points in every iteration. This not only reduces the computation time but also the susceptibility to noise effects (Sculley (2010)).

The classification is performed using a decision tree classifier as structurally illustrated in figure 3, a supervised machine learning algorithm that creates rules based on labeled data sets and allocates every data point in the training data set to the corresponding class based on the rules for each node.

Once the training phase with manually assigned data is completed, the created decision tree is applied to a randomly selected and labeled test data set, which accounts for approximately 30 % of the total data. Based on the test data set, a cross validation is performed in order to evaluate whether the classes created by the clustering algorithm can be identified by the decision tree. If the cross validation results show that an appropriate identification of the defined classes is possible, the created tree is ready to be used

Figure 3: Schematic illustration of the decision tree method for classification.

for unlabeled time series data. The validation of the classification algorithm is performed using the precision and recall, two of the most commonly used measures for machine learning methods. Considering both for imbalanced data sets (classes consist of very different amounts of data points) is important in order to evaluate the ability to picture the accuracy of the predictions made by the classifier. For a detailed explanation of the decision tree method and further supervised as well as unsupervised learning algorithms we refer to James et al. (2013). In addition it is worth to note, that there are more advanced extensions of the simple tree classifier e.g. random forest and bagging tree classifiers that usually demonstrate higher prediction accuracy. But as almost always the case in data science, these more advanced classifiers come at a price, which is the loss of interpretability and transparency (Bühlmann (2012)). Therefore, we implement the simple tree method using the Python library *Scikit-learn* (Buitinck et al. (2013)).

Next, a sensitivity analysis for each individual class is executed to identify the dominant model parameters. Common methods are screening methods, variance based methods and meta-model based methods (Wang and Augenbroe (2017)). According to Menberg et al. (2016) variance based methods provide higher accuracy and hence are suitable for quantitative global sensitivity analysis but also require significantly higher computation time compared to qualitative and local screening methods. As a good compromise between computation time and accuracy we choose the Morris method for analyzing and ranking of model parameters based on the influence they have on the root-mean-squared error (RMSE) between the measured and the simulated output variable. The Morris method is a screening method combined with a factorial sampling design to compensate the disadvantage of derivative-based local methods by scanning the bounded parameter space stepwise (Menberg et al. (2016)). Manually set upper and lower bounds limit the parameter space for each parameter, which is then divided into equidistant parameter variations steps. Each of those describes a point on a trajectory, a vector of parameters which contains the same number of components as the dimension of the parameter space. The number of trajectories as well as the vari-

ation step size Δ_i is defined a priori while the starting point X^* of each trajectory is chosen randomly. Based on these setting parameters, the simulation effort is well known as $t \cdot (p + 1)$ where t is the number of trajectories and p the number of parameters. The appropriate number of trajectories depends on the model complexity. For additional information about sensitivity analysis methods, we suggest the work of Saltelli (2008) and Menberg et al. (2016).

The elementary effects (EE) are used to evaluate the dominance of the parameters. This measure describes the influence of a parameter variation Δ_i on the output y for the i -th component of the parameter vector x . Equation 1 represents the formula for calculating an EE of the i -th parameter. The sensitivity index used to rank the parameters is the absolute arithmetic mean of the EE for every trajectory. Besides the arithmetic mean, a commonly used measure is the standard deviation, which indicates non-linearity and higher order effects. The parameter rankings are forwarded to the optimization stage. The sensitivity analysis is implemented by using the Python library *SALib* (Herman and Usher (2017)).

$$EE_i = \frac{y(\vec{x} + e_i \cdot \Delta_i) - y(\vec{x})}{\Delta_i} \quad (1)$$

The third step of the sequential calibration method is the optimization in which a cost function is iteratively minimized by varying the parameter values. For this cost function, we apply again the RMSE according to eq. 2). Where $\vec{x}$ represents the parameter vector, s_i the simulation output and m_i the measured output of the i -th time step. The advantage taking the RMSE as a cost function is the preservation of the physical unit, which makes it easier to interpret the model discrepancy of the output quantity.

$$RMSE(\vec{x}) = \sqrt{\frac{1}{n} \sum_{i=1}^n (s_i(\vec{x}) - m_i)^2} \quad (2)$$

The particularity about the optimization here is that it is executed separately for each individual classes with the reduced number of parameters. The most dominant parameters for each class are declared as variables while the less influential ones are held constant. In other words, each parameter is optimized in the respective class, in which it shows the highest sensitivity index. By this means, we ensure that every parameter is only optimized once. Non-influential parameters neglected for the optimization. For fine tuning of parameters that show influence in more than one class, after the class individual optimization we execute a final global optimization for all data points jointly with a significantly reduced step size. The order of classes for the optimization are chosen in a way, such that the one with the most dominant parameters is optimized first. This is important because the

Figure 4: Interaction between sensitivity analysis and optimization for two exemplary classes (Esmailzadeh (2018)).

subsequent optimization procedures change a subset of parameters, and thus, also the previous results for the already optimized classes. By following this order, we keep the impact of change at a minimum. The interaction between the sensitivity analysis and the optimization is illustrated schematically in figure 4.

The challenge of non-linear and non-convex optimization is widely known in BES modeling since most of the BES-models contain highly non-linear equations and thus are difficult to optimize. The proposed calibration method provides a way to improve the optimization results with commonly available open source optimizers in the programming language Python. However, one required feature of the optimizer is the ability to handle simple boundaries and least squared error problems to realize a *RMSE* minimization. For this purpose we use the *scipy.optimize.minimize* function of the Python library *Scipy* (Jones et al. (2001)). According to the mathematical derivation by Kraft (1988) and Byrd et al. (1995) the selectable Quasi-Newton method *SLSQP* and *L-BFGS-B* are well suited for this kind of problems.

Case Study

In order to examine the functionality of the presented method we conduct a case study for that we use time series data gained during an independent hardware-in-the-loop (HiL) experiment of a real heat pump (HP). These kinds of tests are dynamic experiments where a building performance simulation is coupled to an energy conversion system on a test bench (Nürenberg et al. (2017)). Figure 5 shows the schematic overview of the experimental setup. The test bench, consisting of a hydraulic part and a climatic chamber, is capable of emulating highly dynamic boundary conditions in terms of heated and cooled water flows (hydraulic part) and thermally

Figure 5: Overview of experimental setup with focus on the energy conversion system (heat pump + condensing gas boiler + combi tank).

conditioned air in the climatic chamber, e.g. to emulate outdoor air of a winter day.

The used monitoring equipment are sensors for volume flows (water, gas, brine), temperatures and electric power. Their installation positions are displayed in figure 5. With a sample rate of 1 s the data set ensures a dense resolution to perform preprocessing analyses in the context of technical systems that need a sophisticated evaluation from a dynamic point of view. The monitoring equipment positioned around the HP unit itself allows energy balances for the electric consumption as well as the provided thermal energy. Furthermore, the gas consumed by the condensing gas boiler can be measured. Unfortunately, there is not enough space between boiler and tank to place a volume flow meter for balancing the thermal gains of only this heat generator. Yet, we are able to determine the thermal demands of both consumer circuits: Space heating and domestic hot water (DHW).

An experimental time frame of about 20 h (73900 samples) is used for the evaluation the presented calibrating method. The inlet and outlet temperature curves as well as the volume flow were already illustrated in figure 2. Besides these physical quantities, we also use the compressor power for performing the classification and additionally the first and the second derivatives of the inlet and outlet temperature for clustering. The derivatives are important in order to capture dynamic effects. Furthermore, a simple moving average filter with a window size of 500 seconds is used for the derivatives to reduce the influence of noise effects on the clustering. As already explained, for classification, the optimization is divided into a training and a test data set to avoid overfitting. Therefore, we use approximately 70 % of the data as training and 30 % as test data for cross validation, which is a common ratio in data science. In contrast to the cross validation for the decision tree classifier, we do not select the training and the test data randomly but ensure that both data sets are representative and contain all kinds of operating phases we identify.

Regarding the temperature curves in figure 2 we can

manually identify three fundamentally different operating procedures that occur repeatedly: A heat-up phase, a heating and a cool-down phase. As one can observe, the heat up phase is comparably short while the cool down phase is clearly dominant in the considered experimental data.

The objective HP model is part of the open-source Modelica library *AixLib* (Müller et al. (2016)), which is available on <https://github.com/RWTH-EBC/AixLib>. The model bases either on mathematical functions or tabulated data according to DIN-EN-14511 (2018). Besides this black box approach for the refrigerant circuit, the rest of the model is grey box. In particular this means that it comes along with thermal capacities, heat losses and pressure drop at both heat exchangers (condenser and evaporator). The thermal output is determined by a variable compressor speed signal. However, since the system controller regulates the particular HP, just with an on/off signal, we apply this control to the simulation model as well. The HP model uses a PT_1 element to be able to represent heat-up and cool-down phases of the compressor. The total model consists of 86 sub-models and a total of 384 scalar equations. For this study we consider the six parameters that describe performance scaling, heating losses and time delays.

Results

As already explained, mainly three classes can be distinguished manually by analyzing the inlet and the outlet temperature curves. Hence, in a first try we set number of centroids to three for the Mini Batch K-means method. The result of the clustering analysis is illustrated in figure 6. We recognize the three clusters found do not correspond to the ones initially expected. Although the Mini Batch K-means assigns over 90 % of measurements correctly, the beginning of the heat-up phase and the heating phase are recognized as only one cluster, whereas the algorithm assigns two clusters to the cool-down phase. The lower one of cooling phases sometimes even passes over into the heat-up phase.

Figure 6: Resulting labeled time series data after clustering with three clusters using Mini Batch K-Means. Colored clusters cannot be mapped to heat-up, heating or cool-down phase.

Increasing the number of centroids to four leads to results illustrated in figure 7. It can be seen that the clustering algorithm now clearly separates the heat-up and the heating phase. Furthermore, the cluster shapes of the formerly identified clusters become

Figure 7: Resulting labeled time series data after clustering with four clusters using Mini Batch K-Means. Colored clusters can be mapped to heat-up (red), heating (green) and two cool down phases (dark and light blue).

sharper and we clearly identify the a priori expected clusters plus one additional within the cooling down phase at lower temperature levels.

Using 70 % of the labeled data set for training of the decision tree classifier, we find that the classifier assigns about 99 % of the measurements of training data correctly. This is also proven by a cross validation.

Since the reference to the automated method is a manual calibration method, we use the three a priori expected clusters for the sensitivity analysis and the optimization, whereas we sum up the two clusters of the cool down phase to only one for better comparability. This assumption is reasonable because, as already mentioned the cool down phase is modeled as only one PT_1 . The sensitivity analysis reveals that six investigated parameters have influence for the heat pump simulation model. Consequently, these parameters are set as variables for the optimization in the respective class, in which the sensitivity index is the highest.

The fitting accuracy is evaluated by using statistical measures that describe the model discrepancy like the RMSE and the coefficient of determination (R^2). Both are presented in table 1.

Table 1: Statistical evaluation of model discrepancy after calibration.

Class	RMSE in K	R^2
heat-up	1.57	0.87
heating	1.03	0.72
cool-down	0.44	1.00
Total	0.65	0.99

Besides the statistical evaluation we use the diagrams shown in figure 8 and figure 9 for visualizing the discrepancy between the simulation and the measurement. We find that the trajectories of simulation and measurements are mostly congruent with only small deviation during the cool-down phase. This is mostly recognizable at the beginning of the cool-down phase. The second diagram also proves the absence of significant simulation errors.

Discussion

The clustering results show that the algorithm is only partly able to create classes, which represent the three physical effects we are able to distinguish manually.

Figure 8: Calibration results illustrated as comparison of simulated data versus measurements for the defined testing data as time series plot.

Figure 9: Calibration results illustrated as comparison of simulated data versus measurements for the defined testing data as scatter plot.

Obviously, the heat-up and the heating phase are not delimited from each other while the cooling phase is divided into two different states. Consequently, the algorithm fails to entirely capture the physical effects entirely due to the similarity of the features for the heat up and the heating phase. However, the algorithm evaluates the difference between the two cooling down stages greater and thus, divides the decline based on the values for the first derivative of the supply and the return temperature. Increasing the number of clusters leads to the desired results, although the clustering algorithm still separates the cool down phase into two parts.

Since the algorithm does only neutrally interpret the provided statistical features without any deeper knowledge about the system, this effect can also be interpreted as lack of detail within simulation models that are not initially expected. Furthermore, this kind of cluster analysis can be further used in order to improve the model and for instance model the cool down as a two stage process or a PT_2 instead of a PT_1 element.

The sensitivity analysis works as expected. Although the Morris method is a rather simple method when compared to more advanced methods it works efficiently and leads to the desired results and enables a mapping of the investigated parameters to the individual classes with low computational effort. Although the method works fine within the scope of the presented use case future work may include the use of the median instead of arithmetic mean for better robustness against singularities. Additionally, for investigation of more complex models incorporating higher number of parameters other methods such as linear regression or Sobol as described by Menberg et al. (2016) may lead to better results. Although the computational effort may increase significantly.

The obtained results of the sequential optimization are promising when compared to non-decomposed calibration. Due to the automated decomposition into sub calibration processes, it was easier to find accurate model parameters for the overall system. Simultaneously, the computation time for the error minimization is shorter because the overall mathematical problem converges faster. However, the results depend on the order the individual suboptimizations are executed. Starting with heat-up provides the best results because the system is in equilibrium with its environment. Nevertheless, this order still needs to be automatized.

We mostly automated the individual steps of the overall calibration algorithm, which enables a comfortable way for the calibration of complex simulation models based on real data. Applying the method to MPC, it will be possible to support the automated implementation process. Furthermore, it will pave the way for adaptive MPC for BES that are able to react to changing boundary conditions of the real systems.

Conclusion

In this work, we presented an approach that enables an efficient and automated calibration of simulation models of BES. It is applicable for online model calibration within model based control algorithms for BEMS and is structured in three modular parts: First, an automated time series analysis consisting of a clustering of operation states using K-Means and an automated classification of measurements using a decision tree classifier. Second, a class specific sensitivity analysis of model parameters. Finally, a multi-stage parameter optimization.

For proof concept, we demonstrated the considered approach, using measurements from a HiL-test of a real heat pump. In a first approach we set the number of clusters to three, where we find that K-means identifies other clusters within the measurements than the three also manually distinguishable clusters that were initially expected. A cross-validation shows that about 76 % of the measurements are clustered as expected. However, adding an additional cluster we find the expected operation phases plus one and significant better identification accuracy. The following classification of measurements shows an accuracy of about 99 %.

The cluster specific applied sensitivity analyses enables a mapping of the parameters to their relevant measurements. Therefore, we are able to split the optimization of model parameters into a multi-stage process, where we only optimize a reduced set of parameters per cluster. Hence, the overall optimization problem is simplified significantly and therefore the chance for convergence increases significantly. This is also proven by results of the demonstration, where we obtain a good accordance of simulation results and

measurements with an RMSE of 0.65 K for the evaluated set of testing data.

Essentially, the algorithm performs as expected although the results may depend on the chosen settings. However, due to its modular design the interim results can be checked after each step, which leads to high level of reliability. Furthermore, it makes the calibration lucid to the user. Additionally, the multi-stage optimization leads to better convergence of the parameter optimization. Compared to a manual calibration the process automation reduces required time to a minimum, which makes the algorithm promising for real world applications for model based control systems. Nevertheless, future work will include better setting estimation and interface automation in between the modules. Also the extension of other sensitivity and optimization methods may be desirable. Furthermore, we will investigate the performance of the calibration for additional HVACR-systems based on possibly incomplete real energy monitoring data from buildings.

Acknowledgement

We gratefully acknowledge the financial support provided by the BMWi (Federal Ministry for Economic Affairs and Energy), promotional reference 0350018A.

Nomenclature

BES	building energy systems
BEMS	building energy management systems
DHW	domestic hot water
EE	elementary effects
HP	heat pump
HVACR	heating ventilation air conditioning and refrigeration
MPC	model predictive control
m_i	i -th measurement
opt	optimized
R^2	coefficient of determination
RMSE	root-mean-squared error
res	result
s_i	i -th simulation output
var	variable
$\vec{x}$	parameter vector

References

- Afram, A. and F. Janabi-Sharifi (2014). Theory and applications of hvac control systems – a review of model predictive control (mpc). *Building and Environment* 72, 343–355.
- Baldi, S., I. Michailidis, C. Ravanis, and E. B. Kosmatopoulos (2015). Model-based and model-free “plug-and-play” building energy efficient control. *Applied Energy* 154, 829–841.
- Béjar, J. (2013). K-means vs mini batch k-means: A comparison.

- Bühlmann, P. (2012). Bagging, boosting and ensemble methods. In J. E. Gentle, W. K. Härdle, and Y. Mori (Eds), *Handbook of Computational Statistics*, pp. 985–1022. Berlin, Heidelberg: Springer Berlin Heidelberg.
- Buitinck, L., G. Louppe, M. Blondel, F. Pedregosa, A. Mueller, O. Grisel, V. Niculae, P. Prettenhofer, A. Gramfort, J. Grobler, R. Layton, J. Vanderplas, A. Joly, B. Holt, and G. Varoquaux (2013). Api design for machine learning software: experiences from the scikit-learn project.
- Byrd, R., P. Lu, J. Nocedal, and C. Zhu (1995, 9). A limited memory algorithm for bound constrained optimization. *SIAM Journal of Scientific Computing* 16, 1190–1208.
- Coakley, D., P. Raftery, and M. Keane (2014, 09). A review of methods to match building energy simulation models to measured data. *Renewable and Sustainable Energy Reviews* 37, 123141.
- DIN-EN-14511 (2018, 05). Air conditioners, liquid chilling packages and heat pumps for space heating and cooling and process chillers, with electrically driven compressors; german version.
- Dittmar, R. and B.-M. Pfeiffer (2006). Modellbasierte prädiktive regelung in der industriellen praxis (industrial application of model predictive control). *at – Automatisierungstechnik* 54 (12), 590–601.
- Eisenhower, B., Z. O'Neill, S. Narayanan, V. A. Fonoberov, and I. Mezi (2012). A methodology for meta-model based optimization in building energy models. *Energy and Buildings* 47, 292 – 301.
- Esmailzadeh, A. (2018). *Automated calibration of simulation models for energy systems supported by machine learning*. Master's thesis, RWTH Aachen University, Aachen.
- Fabrizio, E. and V. Monetti (2015, 04). Methodologies and advancements in the calibration of building energy models. *Energies* 8, 2548–2574.
- Fütterer, J. P., T. P. Schild, and D. Müller (2017). Gebäudeautomationssysteme in der praxis.
- Herman, J. and W. Usher (2017). Salib: An open-source python library for sensitivity analysis. *The Journal of Open Source Software* 2(9), 97.
- James, G., D. Witten, T. Hastie, and R. Tibshirani (2013). *An Introduction to Statistical Learning: with Applications in R*. Springer Texts in Statistics. Springer New York.
- Jones, E., T. Oliphant, P. Peterson, et al. (2001). SciPy: Open source scientific tools for Python. [Online; accessed [today]].
- Killian, M. and M. Kozek (2018). Implementation of cooperative fuzzy model predictive control for an energy-efficient office building. *Energy and Buildings* 158, 1404–1416.
- Kraft, D. (1988). *A Software Package for Sequential Quadratic Programming*. Deutsche Forschungs- und Versuchsanstalt für Luft- und Raumfahrt Köln: Forschungsbericht. Wiss. Berichtswesen d. DFVLR.
- Li, Q., L. Gu, G. Augenbroe, C. Wu, and J. Brown (2015, 12). A generic approach to calibrate building energy models under uncertainty using bayesian inference. In *Proceedings of BS2015: 14th Conference of International Building Performance Simulation Association*.
- Menberg, K., Y. Heo, and R. Choudhary (2016). Sensitivity analysis methods for building energy models: Comparing computational costs and extractable information. *Energy and Buildings* 133, 433 – 445.
- Michailidis, I. T., T. Schild, R. Sangi, P. Michailidis, C. Korkas, J. Fütterer, D. Müller, and E. B. Kosmatopoulos (2018). Energy-efficient hvac management using cooperative, self-trained, control agents: A real-life german building case study. *Applied Energy* 211, 113–125.
- Müller, D., M. Lauster, A. Constantin, M. Fuchs, and P. Remmen (2016). Aixlib - an open-source library within the iea-ebc annex60 framework. In *BauSIM 2016: Sixth German-Austrian IBPSA Conference*, pp. 3–9.
- Nürenberg, M., P. Mehrfeld, K. Huchtemann, and D. Müller (2017). Hardware-in-the-loop test bench setup and its application to determine seasonal performance of heat pump systems. In *12th IEA Heat Pump Conference 2017 Rotterdam*.
- Reddy, A. (2006, 01). Literature review on calibration of building energy simulation programs: Uses, problems, procedure, uncertainty, and tools. *ASHRAE Transactions* 112, 226–240.
- Saltelli, A. (2008). *Global sensitivity analysis: the primer*. John Wiley.
- Sculley, D. (2010). Web-scale k-means clustering. In *WWW 2010: Proceedings of the 19th Annual International World Wide Web Conference*.
- Wang, Q. and G. Augenbroe (2017). Combined sensitivity ranking of input parameters and model forms of building energy simulation. In *Proceedings of BS2017: 15th Conference of International Building Performance Simulation Association*, pp. 3–9.