MULTI-CRITERIA HVAC SYSTEMS OPTIMIZATION WITH MODELICA

Torsten Schwan[1], René Unger[1], Christian Lerche[2], Bernard Bäker[3]
[1]EA Systems Dresden GmbH, Dresden, Germany
[2]Ingenieurbüro Dr.-Ing. Christian Lerche, Pirna, Germany
[3]Institute of Automotive Technologies Dresden (IAD), Dresden, Germany

ABSTRACT

This paper presents a new methodology of machine-aided HVAC system optimization with currently available building simulation and optimization frameworks. The method consists of a two-part approach using Modelica for accurate but simple system simulation as well as Matlab for automated model configuration, result evaluation and parameter variation.

After a short motivation, the paper describes the basic idea as well as the individual algorithm steps of the new method. Further sections describe method-specific requirements to model implementations as well as some important facts about chosen optimization approach. Finally, the paper closes with some sample simulation results.

INTRODUCTION

During the last fifteen years, a wide range of building technologies, renewable energy and new storage concepts have been developed. Together with an overall increase in HVAC (Heating, Ventilation, Air Conditioning) systems efficiency, the fossil fuel consumption and the global warming effects of buildings could be greatly reduced. The disadvantage of these technologies is the growing complexity of the subsystem interaction (e.g. photovoltaic, solar thermal, heat storages, batteries, eMobility, etc.). Additionally, the component control algorithms and system wide energy management algorithms become more and more sophisticated. Obviously, this many degrees of freedom are difficult to handle during the building design phase and require a lot of engineering. More automated approaches are necessarily needed.

Building and energy simulation support the engineer in finding suitable system configurations and solutions for control systems. However, most existing simulation approaches require extensive pre- and post-processing to achieve good results. This paper presents a new approach to better automate model generation and optimization. This helps to achieve suitable results within a significantly reduced time period compared to the current process.

Building construction requires intensive planning, especially the development of a suitable solution for the energy supply. Considering all aspects of current legislation, system costs and environmental friendliness is a huge challenge for architects and HVAC engineers. The wide range of possible solutions for HVAC system configuration and new building construction methods will increase the required planning time in the future, especially for special purpose buildings like hospitals, schools or factories. Today, in such cases, automation can only provide limited assistance.

However, the main overall building energy demand in a country is not caused by such specific projects. Most energy is used by the majority of buildings which are dwelling houses, office blocks, etc. However, the energy saving potential for such a building, in absolute figures, is often small. Therefore also the budget for complicate HVAC planning and system optimization is limited.

Automated optimization approaches will help to improve this situation. Increasing availability of computing capacity (e.g. server farms) as well as new simulation-related solutions for model computing (e.g. Functional Mockup Interface – FMI, 2015) will help building engineers to automatically find the right solution for common tasks within adequate time periods.

During the last five to ten years, building energy simulation has become a part of a planners' everyday-life. Based on first educational and scientific fields of application in the 1970s and 80s, a wide range of simulation platforms and user communities have developed (e.g. TRNSYS, 2013 and EnergyPlus, 2015). Some of them are specialized in building physics, others are mainly focused on HVAC systems engineering.

System simulation approaches have been developed for a wide range of applications since the beginning of commercial computer use. Since the 1990s automotive and aerospace industries have become key drivers in simulation technology development. Together with Matlab/Simulink the versatile, non-causal modelling language Modelica has become one of the most powerful engineering tools in this industrial sector.

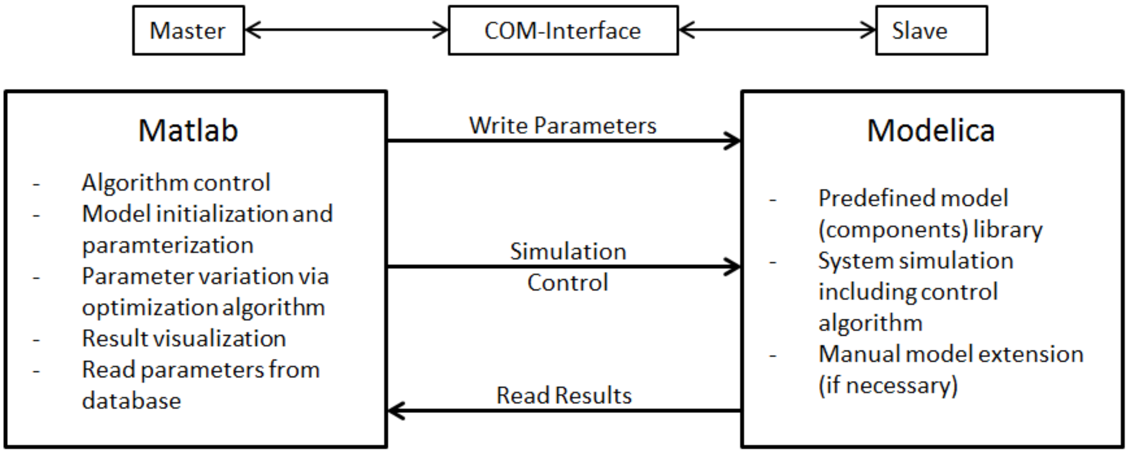

Figure 1: Software structure, individual tasks and data exchange

Since the last five to ten years Modelica has increasingly been used by building engineers and scientists as well. Today, a wide range of different Modelica-based simulation libraries for building physics and HVAC systems simulation are available (Wetter et.al., 2009, Nytsch-Geusen et.al., 2012).

This contribution's methodology uses the Modelica-based tool SimulationX together with the Green Building library and Matlab. This way, Matlab's opportunities in algorithm design and optimization are combined with Modelica's strength in domain-overall physical system modelling and simulation.

Building engineers will be given a toolset to automatically find the right solution for a specific planning task in comparatively short time periods and with high accuracy.

BASIC IDEA AND BOUNDARY CONDITIONS

The idea is to combine the strength of two widely used engineering toolsets to provide an easy-to-handle simulation and optimization framework for building energy systems.

Figure 1 shows the basic structure of the framework under development. In this case Matlab is used as the master tool which controls both optimization cycle and simulation system. The Modelica-based simulation environment is used as the slave tool.

Both tools communicate via COM Interface (i.e. Component Object Model). This includes the specification of model parameters and model preparation as well as simulation control (start, stop, simulation time etc.). Furthermore, relevant simulation results (e.g. energy consumption, etc.) are written via COM Interface in suitable result databases maintained by Matlab.

The shown structure shows the current state of research. After completion of the final proof of concept, it will be implemented as a web-based solution.

For easy usage, parameter dialogues and control applications as well as result visualization will be accessible via web-frontend. Required control and optimization algorithms (currently Matlab) are exported with code generators. Simulation models run as exported FMUs (Co-Simulation) at decentralized servers (Neidhold et.al., 2014). For the future, building engineers will be able to test building energy system layout using the fast-computing and easy-to-understand toolset.

Automation and optimization algorithms can be used best for reoccurring engineering tasks, like dwelling houses or office blocks. This way, presented approach currently focuses on simple building structures. More complex tasks, e.g. district heating grids, complex renewable HVAC systems, etc., can still be solved using existing simulation frameworks (Schwan et.al., 2014) for individual analysis.

Within the last years a wide range of HVAC system components has been developed. They can either provide heat or cold via different media (e.g. ventilation, hydronic systems, etc.) for buildings or internal processes or produce heat/cold and electricity (e.g. CHPs) simultaneously. Furthermore, several different types of energy storages (e.g. batteries) and renewables (e.g. solar collectors) extend this huge number of system types.

The current approach handles a set of comparatively simple system configurations for heat and electricity supply including appropriate storages and renewables:

- Heat Pump
- Boiler
- Combined Heat and Power Unit
- Solar Thermal
- Photovoltaic
- Battery
- Heat Storage Tank
- District Heating

The main goal of optimization is to find the best heat and electricity supply system configuration for a given building regarding efficiency and lifecycle cost. In a configuration, some of the considered components are essential and some of them only provide further opportunities regarding energy and cost efficiency. An analyzed building at least needs one heat supply system, like heat pump or district heat. However, an optimal solution can consist of a multivalent system configuration as well. This way, Table 1 shows suitable combinations of different heat supply systems, which can be analyzed with this approach.

Table 1: Suitable combinations of heat supply systems

	Heat Pump	Boiler	CHP	Solar Thermal	District Heating
Heat Pump	x	✔	x	✔	x
Boiler	✔	x	✔	✔	x
CHP	x	✔	x	✔	✔
Solar Thermal	✔	✔	✔	x	✔
District Heating	x	x	✔	✔	x

Additional renewables and storage systems, like heat storage, batteries and photovoltaics, help to reduce overall electricity consumption or improve renewable energy share. They are not necessarily needed and will be added to different configurations within the optimization algorithm.

The initial implementation of presented approach is limited to the previously shown components and a maximum of two different heat supply systems in one analyzed system configuration. This limits maximum number of analyzed configurations to 28.

These assumptions are valid because system costs tremendously increase for more complex configurations. Simple buildings, like dwelling houses, usually do not have the energy saving potential to justify these costs. However, presented approach can also be adapted to further components (e.g. cooling and ventilation systems) or more complex configurations (e.g. heating system with several temperature levels).

ALGORITHM DESCRIPTION

The goal is to find the best energy system configuration for a specific task regarding different influencing effects, e.g. costs, renewable energy share, system efficiency and energy consumption. Basically, the overall approach consists of four phases (c.f. Figure 2): building definition (1), model initialization (2), iterative simulation (3) and automated evaluation (4).

The first phase includes the implementation of required building physics and usage model. Building and user information (e.g. building standard, heated surface are, number of inhabitants, etc.) are specified manually via a simple input dialogue or automatically by ifc-file, etc. The input dialogue limits the number of parameters to a maximum of 10 to 20.

Further required parameters are added to the building and usage model by suitable statistics (e.g. wall parameters depending on building standard) and integrated schedules for persons' behavior (e.g. electrical energy consumption depending on number of inhabitants) (IWU, 2005). Parameters and statistic values are collected in suitable data bases.

All parameters are used to configure a building model which consists of a maximum of three thermal zones (roof, basement and living/working quarters) (c.f. Schwan et.al., 2014). A building model is automatically created via COM-script commands by referencing the developed Modelica library components. The model is configured with all necessary database values via COM-script as well. This process is limited to a predefined set of available model components (c.f. Table 1) and parameter sets. Adding additional parameters and model components currently requires greater effort and should be avoided.

The simulation of the created building model results the time-variant characteristic of building's heat load and overall electrical energy consumption. These results are then post-processed in Matlab to define HVAC system requirements (e.g. Heat Power). Furthermore, this data defines heat and electricity load schedule in the combined simulation of building and HVAC system in phase 3.

During initialization phase (c.f. phase 2 in Figure 2) the algorithm configures all possible system configurations (max. 28) and checks if some configurations are negligible, e.g. the combination of heat pumps and high flow temperatures (refurbishment of old building). This way, the number of analyzed system configurations and thus required optimization time can be significantly reduced. Again, the HVAC system models are automatically created via COM-script commands. Therefore, the algorithm uses further predefined Modelica models, like heat pump, boiler, etc., including all controls and measurements as well as standardized interfaces.

The third algorithm phase represents the actual optimization cycle. The complete cycle is split into two parts, an initial simulation run of all suitable system configurations (index i – max. 28 – c.f. Figure 2) and the subsequent optimization cycle.

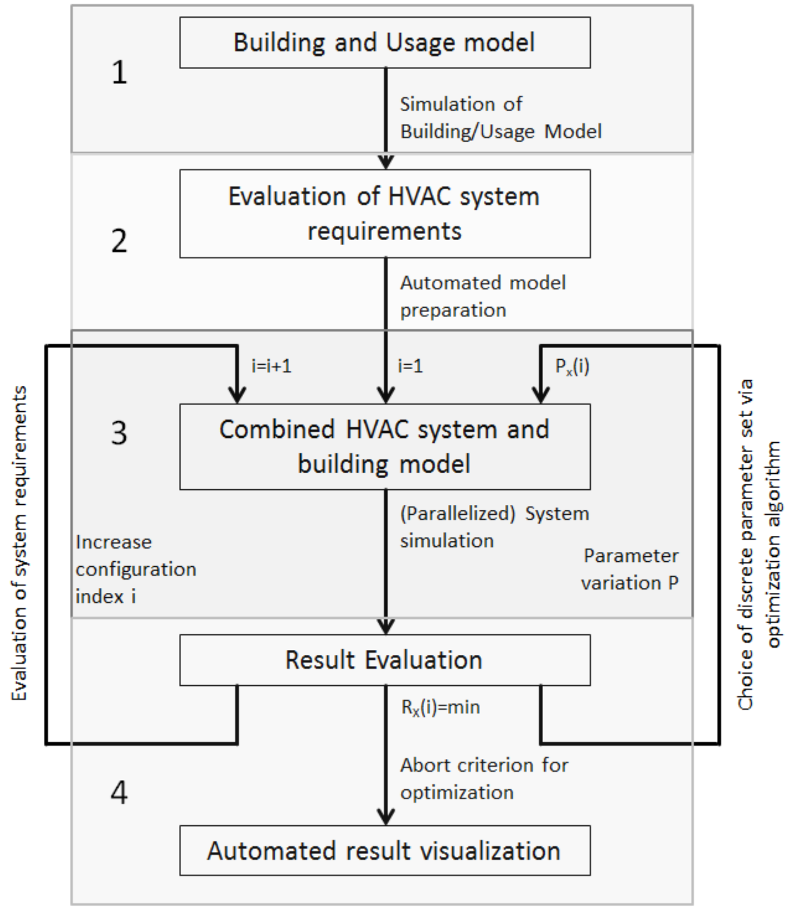

Figure 2: Basic optimization algorithm

All simulation results are evaluated using common optimization criteria in the first part. Configurations which exceed specific disqualifiers (e.g. 20% more running costs and high investments versus mean value of all other components) are rejected. This way, the optimization cycle complexity is reduced.

The second part includes the optimization-based parameter variation for all remaining system configurations. Each system configuration has a specific set of variable parameters (e.g. heat pump power output, heat pump type – air, water, geothermal – c.f. parameter set P_x of configuration index i in Figure 2). These parameters are discretized (if possible). To minimize the number of optimization steps with respect to required optimization accuracy all relevant parameters are discretized using 1-2-5-series. This means that numerical parameters (e.g. size of CHP) are discretized with 1-2-5 values in each decade (e.g. 10 kW, 20 kW and 50 kW), a valid approach to reduce the number of possible parameter variations and this way optimization time.

To find an optimal solution all relevant simulation results are evaluated after each simulation run regarding different optimization criteria (e.g. running costs, overall energy consumption, etc.). The fourth phase includes this evaluation as well as the result visualization (e.g. comparison of accumulated costs).

The optimization cycle actually consists of two mutually dependent cycles. On the one hand, the optimization algorithm iterates all still remaining system configurations ($i < i_{max}$). Parameter variation is done in parallel regarding current simulation and optimization results. If one analyzed system configuration fails the disqualifiers, the number of relevant configurations decrements. The overall algorithm stops if all remaining system configurations ($i = i_{max}$) with all possible parameters are analyzed or if one or several optimal solutions are found.

MODELING PARADIGMS

The current approach uses detailed Modelica building and HVAC systems library models.

An adequate optimization process requires a huge number of simulation runs. Furthermore, suitable simulation models need hundreds of parameters to perfectly match the desired system behavior. Therefore, model configuration uses reduced and easy-to-collect parameter sets, which are available from standard system-specific data sheets (e.g. MPP voltage characteristics of photovoltaic system). Remaining parameters (e.g. temperature-specific efficiency characteristics of heat pumps) are replaced by results of statistical analysis. Extensive research was done to identify obligatory system parameters and suitable statistical substitutes.

The automatic model generation via COM-Interface and Matlab prefers a suitable set of predefined model components with few consistent and standardized

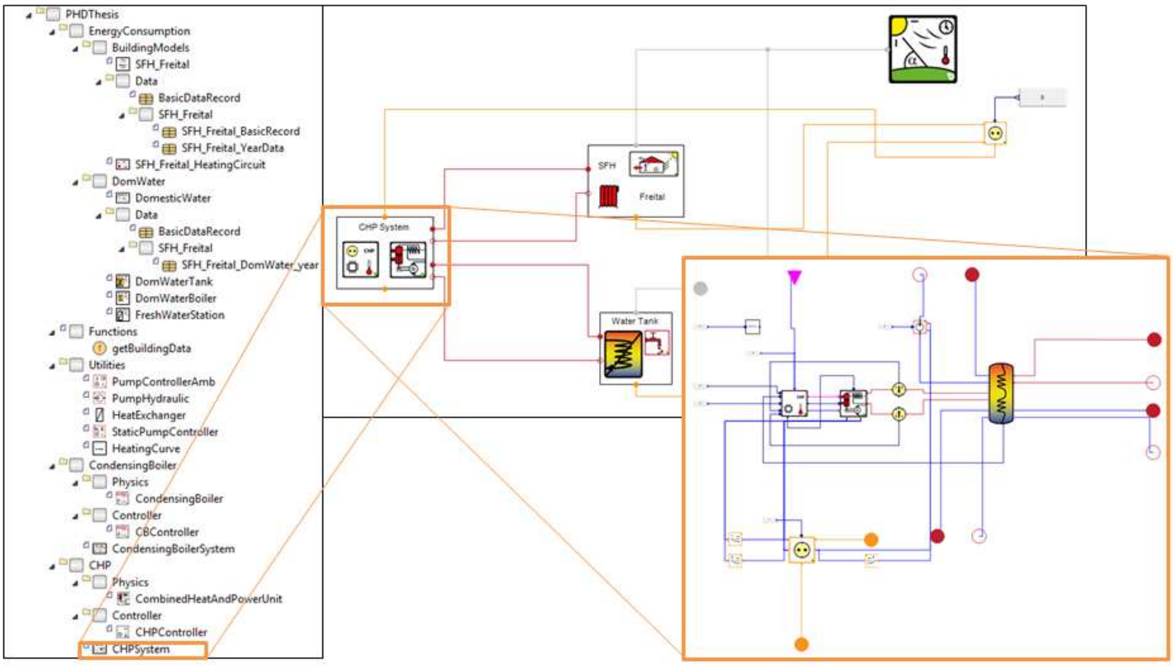

Figure 3: Example model with CHP, building, domestic water and environment unit

interface devices. This is important because the automation process uses predefined model patterns to automatically assemble the final simulation model.

To cope all these challenges a new Modelica-based simulation library was derived from already existing solutions. This library includes all required pre-assembled model components, e.g. for CHP, boiler or heat pump systems.

Figure 3 shows a part of the implemented model library. The exemplary assembling of one HVAC system model (simple CHP) is presented as well with load models for heating, domestic water and electricity consumption. This model equals one of the possible system configurations which are automatically built after second phase of the algorithm in Figure 2.

Basically, all built models consist of an environment model (weather data), heat and electricity consumption, and, if required, domestic water consumption as well as the desired HVAC system. Heating and electricity load are pre-calculated in the first phase of the algorithm. This way, simulation time of each HVAC system model can significantly be reduced because mutual influences between building and HVAC system control are neglected.

Obviously, the number of required interfaces of such a simplified HVAC system model is comparatively low. At least the model only needs interfaces for heat (flow and return connections for space heating and for domestic water supply) and electricity. These interfaces are directly connected to the load models for building heat and domestic water as well as electricity consumption.

With this consistent set of interfaces and parameter dialogues all required HVAC system models can be automatically created using SimulationX-specific COM-script commands and object properties (i.e. Connections, Parameters, etc.).

However, each component model represents different system functionality. The basic component model layout includes system physics, suitable control algorithms, and, if required, further storage specifications (c.f. CHP with heat storage in Figure 3). The specific component structure depends on the final model parameterization and is characterized during the automatic assembling of each system configuration model (blue lines in Figure 3 show variable interface connections). For example a CHP model has optionally influences on domestic water supply. Additional peak load boilers or solar thermal collector can be added as well.

Special emphasis is put on required simulation results for automated system evaluation. Because of a great variety of financial aid for renewable energy production and cogeneration, for example combined CHP and photovoltaic systems require up to four different data points (e.g. annual electricity production CHP, PV) for system evaluation.

Another focus is set on the implementation of additional storage systems, especially heat storage. Basically, heat storage systems are required to buffer and to decouple heat between production and consumption. To reduce library size as well as number of possible system configurations, heat storages are directly included in the relevant HVAC system models (c.f. Figure 3). In case of a multivalent HVAC system model internal heat storage systems can be neglected or several components (e.g. solar thermal collector and CHP) can use the same storage, if required. Such configurations can easily be parameterized during the automatic model assembling as well.

PRINCIPLES OF OPTIMIZATION

System optimization always requires extensive analysis, especially for building systems with a wide range of influencing characteristics and possible solution. Some of them even have negative influence on some of the others, e.g. highly energy-efficient system configurations, mostly cause expensive investments. This is obviously a prototype of a multi-criteria optimization problem.

Lots of criteria can be analyzed during a suitable optimization cycle. However, presented approach is limited to most important ones:

- Investment costs
- Running costs
- Ecological footprint
- Renewable energy share
- System efficiency

From customers' point of view these criteria include the most important information about an optimal system configuration. Further aspects, like degree of self-sufficiency or utilization rate, could be added but are neglected regarding optimization time. Furthermore, the optimization approach has to respect the human level of comfort. Simple optimization approaches might achieve minimum heat power demand with inappropriate levels of indoor temperature or humidity. This is not acceptable. That presented approach ensures the required indoor temperature level during initial building physics simulation phase (c.f. phase 1 - Figure 2).

During the optimization cycle (c.f. Figure 2 – phase 4) these criteria are mainly used to independently evaluate any simulated system configuration and parameter set. Required algorithms (e.g. calculation of running costs) are implemented in Matlab. This toolset already provides adequate toolboxes and functionality to use different optimization methods (i.e. Optimization Toolbox). It is predestinated for algorithm development. Furthermore, the overall optimization time can be reduced if non-physical calculations (e.g. running costs) with no direct influence on simulated system behavior are neglected in the multi-physical simulation.

After each simulation run an adequate optimization algorithm has to decide if the simulated system configuration can be neglected or not. If the configuration is promising, the optimization approach has to vary relevant system parameters to find an optimal system configuration. Therefore, different optimization methods can be used:

- Brute Force Optimization with parameter discretization
- Evolutionary Optimization
- Dynamic Programming
- Particle-Swarm-Optimization

The simple method is brute force. Each possible system configuration is simulated with all possible parameter sets (Saurabh, 2008). The best solution regarding all considered optimization criteria represents the desired optimum. To reduce number of simulation runs the set of analyzed system parameters is discretized in a suitable way (c.f. 1-2-5-series).

However, this method requires the longest time period because of the high number of necessary simulation runs. Each run needs around 5 to 30 minutes for a one year simulation. Therefore reducing the number of required simulation runs is important.

Evolutionary optimization algorithms define an initial set of parameters (initial population) for a specific system configuration (c.f. phase 2 in Figure 2). If no optimum is found, a new set of parameters will be generated using selection, recombination and mutation (c.f. biological processes). The same system configuration is analyzed with varied sets of parameters (populations) until an optimal solution is found (Zitzler, 1999).

Further optimization methods are feasible as well. Some of them are suitable for system control optimization (c.f. Dynamic Programming). Others are preferred for parameter optimization (Particle Swarm Optimization). Because of suitable results regarding degree of optimization and calculation time, this approach uses evolutionary algorithms.

An adequate evolutionary optimization algorithm requires a suitable fitness function. This fitness function is the main result of system evaluation after each simulation run (c.f. phase 4 in Figure 2).

The fitness function represents all relevant optimization criteria. However, different evaluation results cannot be added to a certain fitness value in a simple way because of nonlinearity, different units, magnitudes, etc.

Figure 4 shows the developed geometrical fitness function approach. It mainly combines different fitness values of each criterion to one single optimization result. Each optimization criterion is assigned to one of the axis R_n. These axes describe individual costs for each criterion. The approach is not limited to a certain number of criteria. But the angles between neighboring axes must be all equal and axis values must be positive.

$$\alpha_{n-1,n}=2\pi/n=const \quad (1)$$

$$L_n>0 \quad (2)$$

$$w_n>0 \quad (3)$$

Most customers are only focused on overall system costs. However, others are focused on ecological footprint. This way, individual weighting factors (c.f. w_n in Figure 4) characterize the importance of each optimization criterion.

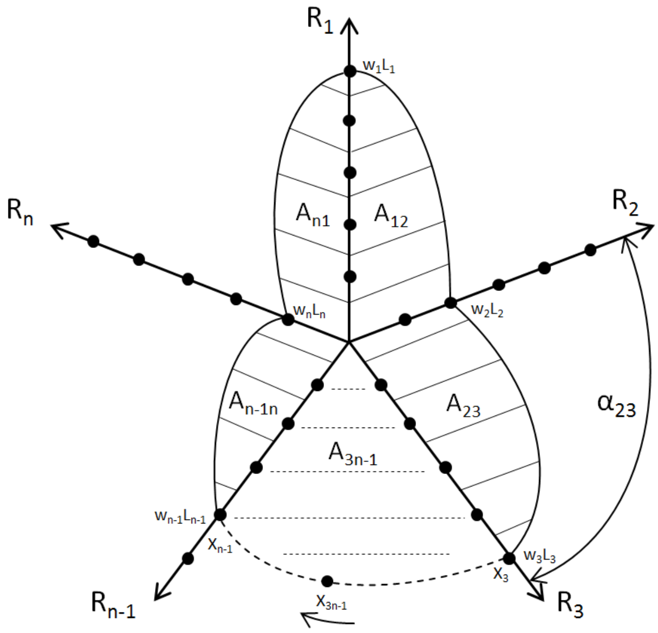

Figure 4: Basic idea of geometrical approximation of multi-criteria optimization problem

Most of the shown optimization criteria always have positive results. In case of negative ones (e.g. ecological footprint: higher grid feed than consumption) the suitable offsets are added. If there are some criteria with high “good”-level and low “bad”-level values (e.g. optimal system efficiency is 100%) related result values (L) are transposed.

The developed fitness function is calculated as the resulting surface area between neighboring axes. These surface areas describe a circular section with a continuously changing radius depending on both axes radii (4):

$$r_{n-1,n} = r_{n-1} + (r_n - r_{n-1}) \cdot \frac{(X_{n-1,n} - X_{n-1})}{(X_n - X_{n-1})} \quad (4)$$

$$r_n = w_n \cdot L_n \quad (5)$$

The axes’ radii result from the product of each criterion’s weighting factor and the individual evaluation result. The required fitness function describes the sum of all surface areas between all neighboring axes.

$$A_{n-1,n} = \frac{\alpha}{8} \cdot (r_n + r_{n-1})^2 \quad (6)$$

$$\sum_{i=1}^{n} r_i = const \quad (7)$$

$$\prod_{i=1}^{n} r_i = const \quad (8)$$

$$F = \sum_{i=1}^{n} A_i = min \quad (9)$$

This fitness function achieves an optimal solution if the sum of all partial surface areas is a minimum. However, this is a multi-criteria optimization approach. Local optima of single criteria can result different optimal fitness function solutions. These results can be compared with a global optimum using Pareto-charts.

Each criterion can be assigned to any axis. The number of analyzable criteria is not limited. If one criterion shall be neglected, the number of axes can be reduced or related weighting factor can be set to zero.

SAMPLE OPTIMIZATION RESULTS

The presented approach is still under development. First tests show replicable and encouraging results.

The basic test case is a single-family house in Germany. This building is fully equipped with smart metering devices for electricity, heat and water consumption.

Building and HVAC system (three-storey building with oil-fired condensing boiler) were modeled using Modelica and the developed model library. First results show a suitable deviation between measurements and simulation of 5-10%:

Table 2: Sample results of model validation

	Measurement	*Simulation*	*Dev.*
Electricity consumption	7.000 kWh/a	7.300 kWh/a	4.2%
Fossil fuel consumption	24.200 kWh/a	22.500 kWh/a	7.3%

The first implemented optimization strategies were tested as well as the resulting model accuracy as well. Therefore, Figure 5 shows the generated outputs of overall system costs calculation (i.e. investment costs and accumulated running costs in relation to existing boiler system (CB) in 20 years). This way, the relevant result of the cost criterion is the amortization time. Low time periods are preferred by the optimization approach.

Depending on overall system costs an optimal system configuration would consist of a geothermal heat pump (HP), partly supplied by a south-faced photovoltaic system (PV2). Other system components, e.g. solar thermal collectors (ST), would increase system costs in an inacceptable way.

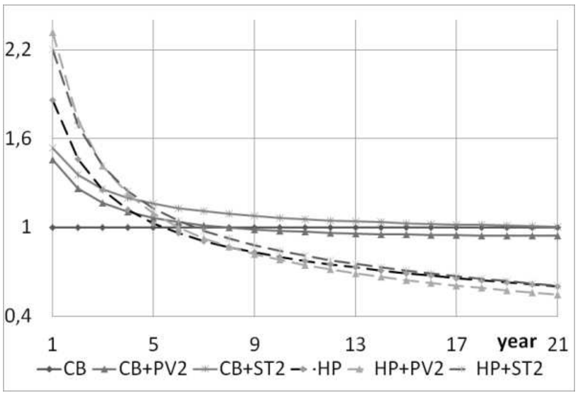

Figure 5: Relative accumulated costs of different HVAC system configurations

For the specific test case, other optimization criteria support the same optimal solution, e.g. heat pump and PV system cause the lowest ecological footprint because of high system efficiency and local electricity production.

CONCLUSION

The presented method combines the power of Modelica in physical systems simulation and the strength of Matlab in optimization and control algorithms design. The approach helps to find an optimal HVAC system configuration with optimal components parameters for yet simple but often occurring building samples. The presented method can be extended as an online platform for simple renewable HVAC systems layout.

Further work is focused on improvement of model accuracy and simulation speed. The new FMI standard is a great help for combining models with different time constants (building physics, HVAC system control) using individually specialized solvers. Furthermore, simulation time can significantly be reduced if required simulation runs can be parallelized. FMI enables the export of simulation models with solver and makes parallel, cloud-based simulation possible (Neidhold et.al., 2014).

ACKNOWLEDGEMENT

The development of presented approach is partly encouraged by subsidies from German Federal Ministry of Economics and Technology.

NOMENCLATURE

A	= Surface are
w	= weighting factor
L	= Result value of specific criteria (length)
R	= Criterion axis (radii equal costs)
r	= Radius of circular sector
α	= Angle between criteria axes
X	= Point of circular arc
F	= Fitness function

REFERENCES

EnergyPlus, 2015: Homepage of DOE, http://apps1.eere.energy.gov/buildings/energyplus/.

FMI, 2015: Homepage of Modelica Association, FMI Standard 2.0.

IWU, 2005: Development of a simplified, statistically proven procedure for survey of building data and compilation of energy performance statistics of buildings, Scientific Project Report, Institut Wohnen und Umwelt GmbH, Darmstadt, Germany.

Neidhold, T., Bittner, S., Oelsner, O., 2014. SimulationX goes online – A Web Platform for Cloud-Based Simulation, 17th ITI Symposium, Dresden, Germany.

Nytsch-Geusen, C.; Huber, J.; Ljubijunkic, M; Rädler, J., 2012: Modelica-BuildingSystems – Eine Modellbibliothek zur Simulation komplexer energietechnischer Gebäudesysteme, BauSIM 2012, Berlin, Germany.

Saurabh, S., 2008: Exact algorithms for optimization and parameterized versions of some graph theoretic problems, PhD thesis, The Institute of Mathematical Sciences, Chennai, India.

Schwan, T., Unger, R., Lerche, C., Kehrer, C., 2014: Model-Based Design of Integrative Energy Concepts for Building Quarters using Modelica, 10th International Modelica Conference, Lund, Sweden.

Schwan, T., Unger, R., Bäker, B., Mikoleit, B. Kehrer. C., 2011: Optimization of local renewable energy systems using automotive simulation approaches, 12th Conference of International Building Performance Simulation, Sydney, Australia.

SimulationX, 2015: SimulationX-Homepage of ITI GmbH, http://www.simulationx.com.

TRNSYS, 2013: Homepage of TRNSYS, http://sel.me.wisc.edu/trnsys/default.htm, University of Wisconsin, Madison, USA.

Wetter, M., 2009: A Modelica-based model library for building energy and control systems, 11th International IBPSA Conference, Glasgow, Scotland.

Zitzler, E., 1999: Evolutionary algorithms for multi-objective optimization: Methods and Applications, PhD thesis, Swiss Federal Institute of Technology Zurich, Switzerland