9th International Workshop on Simulation for Energy, Sustainable Development & Environment
18th International Multidisciplinary Modeling & Simulation Multiconference

ISSN 2724-0061 ISBN 978-88-85741-67-6 © 2021 The Authors.
doi: 10.46354/i3m.2021.sesde.007

Optimization of Complex Thermally Electrically Coupled Buildings using Genetic Programming to Identify Optimal Energy Flow Controllers

Kathrin Kefer[1,*], Patrick Kefer[2], Roland Hanghofer[4], Markus Stöger[1], Bernd Hofer[1], Michael Affenzeller[3] and Stephan Winkler[3]

[1]Fronius International GmbH, Günter-Fronius-Straße1, Thalheim, A-4600, Austria
[2]Research & Development FH OÖ Forschungs und Entwicklungs GmbH: Research Group ASIC, Ringstraße 43a, Wels, A-4600, Austria
[3]Research & Development FH OÖ Forschungs und Entwicklungs GmbH: Research Group Heuristic and Evolutionary Algorithms Laboratory, Softwarepark 11, Hagenberg, A-4232, Austria
[4]Dynatrace Austria GmbH, Softwarepark 32, Hagenberg, A-4232, Austria

[*]Corresponding author. Email address: kefer.kathrin-maria@fronius.com

Abstract

During the last years, renewable energy sources and their management have become increasingly important to help driving forward the energy transition and slow down the global warming. Current energy management systems are either simple but not optimal or very complex, computationally intensive and optimal. Despite that, they also often focus on the optimization of just the electrical energy flows of buildings so far. This work focuses on the development of a linear model predictive controller as well as heuristic energy flow controllers for optimizing a complex thermally-electrically coupled system. For that, a real world building is modelled in MATLAB Simulink and used for the training process of the heuristic controllers as well as for the evaluation of the different optimizers in simulation with different timespans. It is found that the linear MPC works better than a rule-based self consumption optimization and that the heuristic controllers work significantly better than these two for all evaluation timespans up to 180 days, while they perform significantly worse for 364 days.

Keywords: Energy Management System; Genetic Programming; Symbolic Regression

1. Introduction

In order to drive forward the energy transition and slow down the global warming, renewable energy sources and their management have become increasingly important during the last years. As these renewable energy sources are subject to constantly changing environmental conditions, they cannot produce a continuously stable amount of energy, which results in faster and bigger fluctuations in the low voltage grid. Especially when there is a lot of, e.g. photovoltaic (PV) production during a sunny day resulting in a big amount of feed-in energy or high consumption peaks during cloudy or rainy days with only little PV production, the grid operating reserves need to take a lot of effort to compensate this and keep the grid stable. At the moment, this still works quite well, however it will become increasingly difficult with the increasing number of renewable energy sources built and installed.

This is why energy management systems (EMS), which should use, store and distribute the self-produced renewable energy as efficiently as possible mainly in order to minimize the building's energy costs, also become more and more important. However, currently such EMS are either runtime efficient and easy to use but not optimal, like simple rule-based energy management systems. Or they are not real-time capable due to being computationally very intensive while providing at least almost optimal results, like model predictive controls. Despite that, they also often focus on the optimization and management of just the electrical part of buildings so far. Therefore, the aim of this work is to develop a computationally efficient and near-optimal energy management system for a complex thermally-electrically coupled system. This system is modelled in MATLAB Simulink and includes a PV power plant, a hydroelectric power plant, a battery storage, a heat pump, a Fronius Ohmpilot which turns electricity into hot water and an oil heating as primary heating source. The main contributions are the following:

1. Development of a complex thermally-electrically coupled simulation model using a real world building as basis.
2. Learning the optimal structure and behaviour of heuristic energy management controllers using an existing optimization approach, historical system data and genetic programming.
3. Developing a linear model predictive controller as reference optimizer to the heuristic controllers.
4. Optimizing the thermal and electrical energy flows of the system and therefore minimizing its energy costs in simulation using the two presented optimization approaches and one existing EMS.
5. Detailed analysis of the different energy flow controllers in simulation for their ability to minimize the system's energy costs for different evaluation timespans.

The remaining work is structured as follows: Chapter 2 gives an overview of existing EMS technologies, followed by a detailed description of the developed methodology, the used data basis and evaluation procedure in chapter 3. Finally, chapters 4 and 5 describe the evaluation results and the conclusions drawn from them.

2. State of the art

Currently, three main trends can be identified for the optimization of residential energy flows: Rule-based control systems, model predictive controls and also meta-heuristic optimization algorithms as they are used in this work. All approaches are briefly explained below.

2.1. Rule-based Energy Management Systems

Rule-based energy management systems are one of the simplest to optimize energy flows. Experts define rules that should manage the system and represent them as simple either-or decisions, which form a tree structure in the program flow. In these trees, the branches represent the decisions according to the defined rules while the leaf nodes define the actions that the system should perform in the respective state. Rule-based EMS are fast and easy to develop for simple systems and also achieve good optimization results for them. However, optimizing complex systems with this approach requires a lot of expert knowledge and development effort and is prone to unexpected and unwanted side effects. Nevertheless, due to the simple tree structure the execution of such EMS, which is basically just running through the tree from top to bottom, is very performant and can control also larger systems in realtime. Examples for rule-based energy management systems were developed by De Coninck et al. in 2014 (De Coninck et al., 2014), Salpakari and Lund in 2016 (Salpakari and Lund, 2016) and Alimohammadisagvand et al. in 2018 (Alimohammadisagvand et al., 2018). One rule-based EMS that is already sold to customers is the Fronius self consumption optimization (GmbH, a), which is also used as a reference EMS for this work. It is integrated in the Fronius inverters and uses a zero-feed in strategy to self-consume as much of the produced energy as possible instead of feeding it into the grid. As this work aims at optimizing a complex thermally-electrically coupled system, rule-based EMS are likely to not work very well. Additionally, as soon as one part of the system changes, the whole optimization needs to be adapted. With the approach proposed in this work, only a retraining of the controllers is necessary. However, they are similarly fast during the execution independently of the size and complexity of the system.

2.2. Model Predictive Controls for Energy Management Systems

Another widely used but more complex technique for energy management systems are model predictive controls (MPCs). They are linear or quadratic optimization programs that use accurate forecasts and an exact representation of the system to be optimized as a simulation model in order to calculate the optimal control inputs for the next point in time at a given time. For this purpose, the current system values and the simulation model are used to predict the future system behaviour. With that, the optimization algorithm calculates the actions for the next point in time (Kothare et al., 1996). Due to that, such controllers run almost optimally but not in realtime like the approach presented in this work as simulating the model to calculate the forecasts usually takes quite some time. Additionally, MPCs use the simulation model to calculate the forecasts whereas this work uses the model for the training process of the controllers and not during their execution. MPC based energy management systems were developed by Chen et al. in 2012, who optimize the schedule of thermal and non-thermal appliances of residential buildings with their MPC (Chen et al., 2013), in 2018 by Godina et al. whose MPC optimizes and controls the air conditioning of a room within a house that also has a PV system as renewable energy source integrated (Godina et al., 2018) and in 2020 by Seal, Boulet and Dehkordi, who implemented a centralized MPC for a zone based comfort and energy management in a residential building with a PV system, a battery and a heat pump (Seal et al., 2020).

2.3. Meta-Heuristic Optimization Algorithms for Energy Management Systems

Especially in recent years, there has also been a tendency to use meta-heuristic algorithms for controlling energy flows, mostly particle swarm optimization approaches (PSO) and genetic algorithms. For finding optimal solutions, PSO algorithms create a certain number of "particles" (solution candidates) which together form a swarm that moves through the defined solution space (Pedrasa et al., 2010). Examples of PSO based energy management systems were published by Eseye et al. in 2016, who optimize an energy management system for an isolated industrial microgrid for minimal energy costs and maximum economical benefit using a modified particle swarm optimization algorithm (Eseye et al., 2016) and by Sisodiya, Kumbhar and Alam in 2018, who presented a PSO algorithm that schedules a building's electric vehicle, electric water heater, heating, ventilation and air conditioning under specified user requirements in order to minimize the electricity bill (Sisodiya et al., 2018). In comparison to the approach used in this work, Eseye et al. do not train the heuristic controllers in advance but use the PSO to create the optimal schedule directly during the execution of the energy management system but, like this work, also use a simulation model for that. Despite that, their focus is on the optimization of an isolated micro grid system whereas this work focuses on the optimization of residential buildings. Pedrasa et al. (Pedrasa et al., 2010), however, use a more user-focused approach where they let the user decide on the priorities of the building's appliances, which this work does not take into account.

Differing to the PSO algorithms, the genetic algorithms have the natural, biological selection process as paradigm. They generate new solution candidates based on a parent generation using crossover and random mutation. After a (quality) selection, a number of these children solution candidates are transferred to the next generation. In this way, an optimal solution can be approximated or even found in the course of the generations (Srinivas and Patnaik, 1994). One example of genetic algorithms that are used for the optimization or management of energy sources and loads were presented in 2009 by Morganti et al. (Morganti et al., 2009), in 2013 by Arabali et al. (Arabali et al., 2012) and in 2018 by Gonçalves et al. (Gonçalves et al., 2018). Morganti et al. use an agent-based optimization problem representation where each appliance is modelled as an agent in a system and optimize this system in simulation using single and multi-objective genetic algorithms (Morganti et al., 2009). In comparison to that, this work does not use an agent-based problem representation but a symbolic regression problem and a physical simulation model that should be optimized. Besides that, they also use the NSGA-II as multi-objective algorithm but the classic single objective GA instead of the Offspring Selection Genetic Algorithm that is used in this work. Similar to this work, Arabali et al. use a genetic algorithm-based approach to optimize a system with controllable heating, ventilation and air conditioning loads which are supplied by a hybrid-renewable generation and energy storage system. Different to this approach, they use historical data to stochastically model the load, PV system and wind production. Despite that, they also do not use a simulation model for the optimizations but a probabilistic modelling method of the energy production and loads (Arabali et al., 2012). Gonçalves et al. focus on the energy cost minimization of a residential energy resource while considering a set of user defined comfort preferences. In order to optimize these conflicting objectives, they also use a further development of the well-known NSGA-II similar to this work. However, they do not use a symbolic regression based problem representation and a simulation model for the optimization as done in this work (Gonçalves et al., 2018).

3. Method

In the course of this work, two different simulation-based optimization approaches for a complex thermal electrically coupled system were developed. The needed detailed simulation model for that is explained together with the parameters that are optimized in more detail in section 3.1. The developed genetic programming based optimization approach as well as the implemented linear model predictive control, which serves also as a comparison reference for the evaluation of the heuristic controllers, are explained in more detail in sections 3.2 and 3.3. These sections are followed by a detailed explanation of the data basis which is used for the trainings of the heuristic controllers and the evaluation of the different optimization approaches in section 3.4 as well as by a description of the evaluation procedure itself in section 3.5.

3.1. Simulation Model

The basis for the evaluation model is a real-world building in Upper Austria, which is modelled in MATLAB Simulink to be able to simulate its electrical (Fig. 1) and thermal (Fig. 3) energy flows. The electrical appliances built into the system and their energy flows are shown in figure 2 and include a hydroelectric power plant, whose energy production is abstracted in the model by adding it to the household load, a 1.5 kWp PV system, a 12kWh battery storage and an Ohmpilot device (GmbH, b), which turns electric power into hot water using four heating rods. The inverter is modelled linearly as shown in equation 1, where

- P_{PV} is the power produced by the inverter,
- P_L is the household load plus the power produced by the hydroelectric power plant,
- P_{Grid} is the power fed into or consumed from the grid,
- $P_{Ohmpilot}$ is the power consumed by the heating rods via the Ohmpilot,
- P_{Bat} is the power discharged from the battery,
- P_{toBat} is the power charged into the battery,
- P_{toDC} is the power to the DC node,
- P_{toAC} is the power to the AC node,
- η_{PV} is the efficiency of the inverter at the specific voltage,
- $\eta_{Bat,DC}$ is the efficiency from the battery to the DC node,
- $\eta_{DC,Bat}$ is the efficiency from DC node to the battery,
- $\eta_{DC,AC}$ is the efficiency from the DC node to the AC node,
- $\eta_{AC,DC}$ is the efficiency from the AC node to the DC node.

$$\begin{aligned} P_{PV} \times \eta_{PV} + P_{Bat} \times \eta_{Bat,DC} - \frac{P_{toBat}}{\eta_{DC,Bat}} + P_{toDC} - \frac{P_{toAC}}{\eta_{DC,AC}} = 0 \\ P_{toAC} + P_{Grid} - P_L - \frac{P_{toDC}}{\eta_{AC,DC}} - \sum P_{Ohmpilot} = 0 \end{aligned} \tag{1}$$

Equation 2 shows the calculation of the battery's state of charge (SOC) for each simulation timestep t, where ΔT denotes the simulation interval, i.e. the difference between two timesteps and Cap is the battery's capacity.

$$SOC(t+1) = SOC(t) + P_{toBat}(t)\frac{\Delta T}{Cap} - P_{Bat}(t)\frac{\Delta T}{Cap} \tag{2}$$

The thermal part of the simulation model as shown in figure 3 is modelled using the Carnot 2016b blockset (Juelich, 2018) and consists of the building itself in form of a one-node simulation model together with its space heating, the oil heating and the the hot water boiler. This hot water boiler can be heated up using the oil heating or four heating rods with a maximum power P_{max} of 9kW, which are controlled by the Ohmpilot. As shown in equation 3, these four heating rods can be controlled in different ways: heating rod 1 ($P_{el,1}$) can only be controlled using a variable amount of energy, heating rods 2 and 3 ($P_{el,2,3}$) can be controlled either using a variable amount of energy or by turning them on/off with the maximum possible 9kW and heating rod 4 can only be switched on/off. In the equation, i denotes the heating rod, $y_{E,i}$ is the activation variable for the heating rods that can be switched on and off and $y_{var,i}$ represents the activation variable for the heating rods that can be controlled variably.

$$\begin{aligned} P_{el,i} \leq P_{max} \\ \sum_{i=1}^{3} y_{Var,i} \leq 1 \\ y_{E,i} + y_{Var,i} \leq 1 \text{ where } i = [2,3] \\ P_{el,Var,i} \leq P_{max} \times y_{Var,i} \text{ where } i = [1,3] \end{aligned} \tag{3}$$

As this setup in the real world building is quite special, there is an additional control program needed. It uses the available amount of surplus power from the inverter *P_inv* to switch as many heating rods on with the full amount of power and then connects the Ohmpilot to one of the variable controllable heating rods $P_{el,1-3}$ to supply this one with the remaining surplus power *PowerTarget*. This logic is also shown in listing 1, where *P_HeatRod* is an array that stores the amount of power reserved for each heat rod and the *EnableHeatRodsFix* and *EnableHeatRodsVar* variables store the decisions which heating rods should be switched on with the full 9kW or the variable power.

Listing 1. This algorithm is responsible for switching on and off the heating rods with their maximum power and calculating the remaining power target for the Ohmpilot which controls the remaining variable power heating rod.

```
RodIdx = 1;
while P_inv > 0 && RodIdx < 5
  if HeatRodIdx == 1 && P_inv >= Pmax
    P_HeatRod(HeatRodIdx) =  Pmax;
  elseif HeatRodIdx ~= 1
    P_HeatRod(HeatRodIdx) = min(P_inv, Pmax);
  end
  P_inv = P_inv - P_HeatRod(HeatRodIdx);
  HeatRodIdx = HeatRodIdx + 1;
end
EnableHeatRodsFix = (P_HeatRod(1:3) == Pmax);
EnableHeatRodsVar(1:2) = (P_HeatRod(2:3)
```

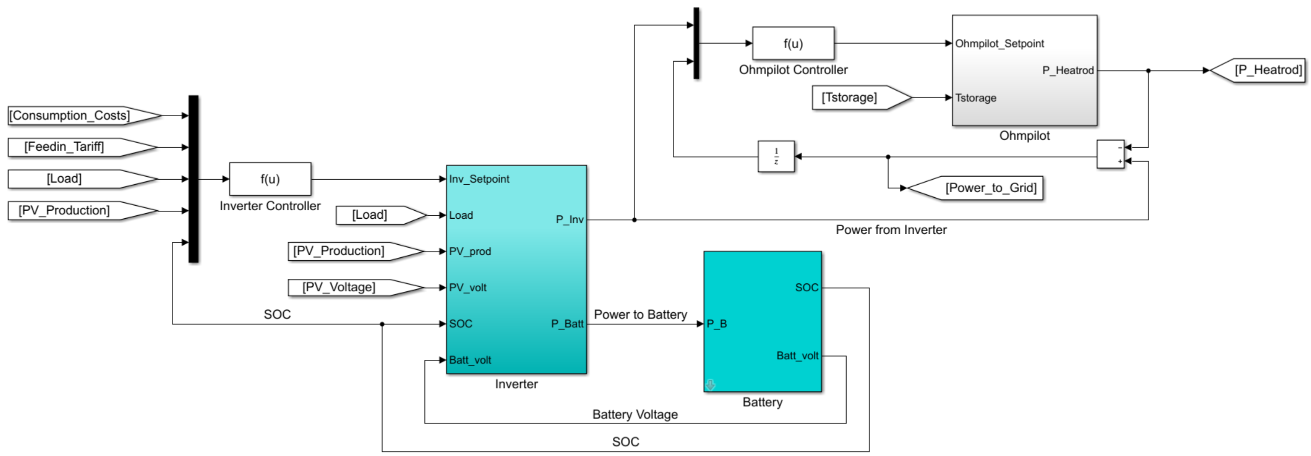


Figure 1. The electrical part of the simulation model with the inverter, the battery and the Ohmpilot. The controller for the inverter gets the energy tariffs, the household load, the production from the PV system and the state of charge of the battery as input parameters to calculate the respective set point. For calculating the Ohmpilot set point, the respective controller uses the currently and remaining available power from the inverter and the energy consumed or fed into the grid at the previous time step. Using the Ohmpilot, the heating rods then heat up the hot water boiler of the system as shown in the thermal part of the system in Figure 3.

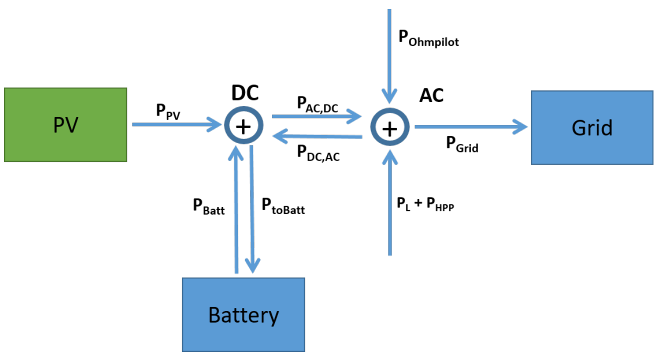


Figure 2. The electrical energy flows of the system between the PV system, the battery, the DC and AC nodes, the household load together with the hydroelectric power plant production, the Ohmpilot and the energy grid.

```
    < Pmax) & (P_HeatRod(2:3) > 0);
EnableHeatRodsVar(3) = P_HeatRod(4) > 0;
if any(P_HeatRod>0)
  PowerTarget = min(P_HeatRod(P_HeatRod>0));
else
  PowerTarget = 0;
end
```

The input values of the model are the energy tariffs for consumption and feed-in, the household load with the hydroelectric power plant energy production added, the energy production and voltage of the PV system and weather data. They are also used by HeuristicLab to optimize three parameters of the model:

- the grid feed-in set point of the inverter using the *Inverter Controller* block,
- the energy available for heating up the hot water boiler using the heating rods and the Ohmpilot using the *Ohmpilot Controller* and
- the enabling or disabling of the return flow mixer using the *Return Flow Mixer Controller*.

The *Inverter Controller* uses the two energy tariffs, the household load with the hydroelectric power plant production included, the PV system production and the current state of charge of the battery to calculate the optimal amount of grid feed-in energy for the inverter (Fig. 1). The still available energy from the inverter is then used together with the amount of energy fed into the grid at the previous simulation step by the *Ohmpilot Controller* to calculate the available power for the heating rods (Fig. 1). Finally, the *Return Flow Mixer Controller* uses the current storage temperature to decide whether to enable or disable the mixer for the hot water boiler (Fig. 3).

The result of the simulation are the total energy costs for the system and simulated timespan. They are calculated as shown in equation 4 where N denotes the total number of simulation steps and t is the current simulation step. There, the amount of energy consumed from and fed into the grid from the *Power_to_grid* signal in Fig. 1 is multiplied with the respective energy tariffs and added to the energy costs caused by the oil heating using the *MoneyFlowOilHeating* signal in figure 3. This is then summed up for all simulation steps to get the total energy costs.

$$costs_{total} = \sum_{t=1}^{N} P_{fromGrid}(t) \times costs_{consumption}(t) - P_{toGrid}(t) \times costs_{feedin}(t) + costs_{backup} \tag{4}$$

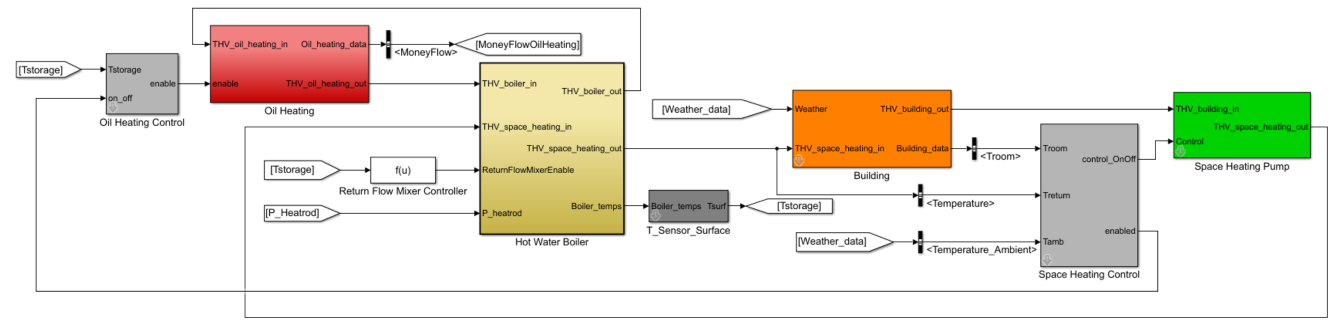


Figure 3. The thermal part of the simulation model gets the power for the heating rods from the Ohmpilot as shown in Figure 1. They heat up the hot water boiler according to that. Despite that, also the oil heating is started by the oil heating control if the boiler temperature drops below a certain threshold. The water from the hot water boiler is then used for the hot water supply in the building and by the space heating control, which decides whether the building needs to and can be heated up or not based on the room temperatures, the boiler temperatures and the outside temperature. For simplicity reasons, this graphic of the model was slightly adapted from the original one, but the blocks and the connections between them were kept the same.

3.2. Genetic Programming based Optimization Approach

For the genetic programming approach, a further development (Kefer et al.) of the model-based energy flow optimization approach developed by Kefer et al. (Kefer et al., 2019) is used. With that, the three parameters of a simulation model described in section 3.1 are optimized to minimize the system's energy costs. This is done using the optimization framework Heuristiclab (Wagner et al., 2010) and MATLAB Simulink (mat, 2020), where HeuristicLab starts the generation of C-code from the MATLAB Simulink model, adapts the generated code with additional functionality and then generates a DLL from that. In the controller training, this DLL is then used to evaluate the different solution candidates by receiving the controllers contained in the solution candidate as a formulas together with the needed input values from HeuristicLab and then being run for a specified number of simulation steps. Once the simulation of the system is done, the energy costs of the system are read from the DLL by HeuristicLab and are used as the quality measurement, based on which the next generation solution candidates are selected (Kefer et al.).

3.3. Linear Model Predictive Control

For the linear model predictive controller, it is chosen to avoid mixed integer variables whenever possible as their usage does neither guarantee convergence nor an optimal result. In addition to that, the computational costs are significantly higher than when using only normal linear programming. The optimization problem for the MPC was defined using a problem based formulation where the optimization matrix is created by the algorithm. A prediction horizon of 24 hours and an optimization interval of 15 minutes are chosen. For that, the thermal storage, the backup oil heating, as well as the building model needed to be modelled in a linear way so that the algorithm can handle their optimization. The storage is modelled as four node finite volume model as there are four heating rods included in the real world system and therefore also the developed simulation model. Using the mathematical formulation of equation 5 for every of these four nodes, the model of the thermal storage is obtained by the optimization algorithm.

$$\begin{aligned} m_n.c_{p,fluid}.T_n = \dot{m}_i n.c_{p,fluid}.(T_{in} - T_n) + \\ \dot{m}_{down}.c_{p,fluid}.(T_{n+1} - T_n) + \dot{m}_{up}.c_{p,fluid}.(T_{n-1} - T_n) + \\ \frac{\lambda}{d_h}.(T_{n+1} - T_n) + \frac{\lambda}{d_h}.(T_{n-1} - T_n) + U_A.(T_n - T_{a,stor}) + \dot{Q}_{ext} \end{aligned} \tag{5}$$

As the heating power of the boiler cannot be controlled by an external control signal, the behaviour of the backup control had to approximately formulated in the optimization algorithm, which resulted in the additional set of equations and constraints shown in equation 6. Despite that, further assumptions like constant mass flow of the backup heating, constant mass flow in the storage and non-existing inverse thermocline are required to get the linear representation of the system.

$$\begin{aligned} &\dot{Q}_{Bck} \leq y_{Bck}\dot{Q}_{Bck} \\ &\dot{Q}_{Bck} \leq 0 \\ &\dot{Q}_{Bck} \leq \dot{m}_{Bck}.c_{p,fluid}.(T_{set,Bck} - T_3) \end{aligned} \tag{6}$$

The building model for the linear optimization approach was modelled using a 3R2C approach (equation 7) similar to the representation in the simulation model. The radiator is modelled with a constant heat transfer coefficient while all the external heat sources are considered as inputs to the linear optimization model and the mass flow of the heating system is considered to be constant.

$$\begin{aligned} cap_{\text{house}}\, \dot{T}_{\text{house}} &= \dot{Q}_{\text{house}} - u_{A,\text{Heat}} \left(T_{\text{house}} - T_{m,\text{heat}}\right) \\ cap_{\text{heat}}\, \dot{T}_{m,\text{heat}} &= u_{A,\text{Heat}} \left(T_{\text{house}} - T_{m,\text{heat}}\right) + \\ & c_{p,f}\, \dot{m}_{\text{heat}} \left(T_{i,\text{heat}} - T_{m,\text{heat}}\right) \end{aligned} \tag{7}$$

The whole thermal model with the building and storage models included can be written as state space system as shown in equation 8, which represents a series of constraints for the optimization problem.

$$\begin{pmatrix} T_{1,\text{k+1}} \\ T_{2,\text{k+1}} \\ T_{3,\text{k+1}} \\ T_{4,\text{k+1}} \\ T_{\text{house,k+1}} \\ T_{m,\text{heat,k+1}} \end{pmatrix} = A_k \begin{pmatrix} T_{1,k} \\ T_{2,k} \\ T_{3,k} \\ T_{4,k} \\ T_{\text{house},k} \\ T_{m,\text{heat},k} \end{pmatrix} + B_k \begin{pmatrix} P_{\text{el1},k} \\ P_{\text{el2},k} \\ P_{\text{el3},k} \\ P_{\text{el4},k} \\ \dot{Q}_{\text{bck},k} \\ \dot{Q}_{h,\text{ext},k} \\ T_{\text{amb},k} \end{pmatrix} \tag{8}$$

Due to the thermal mass of the system, the forward temperature and the storage temperature also have to be set as soft constraints as shown in equation 9 to guarantee the convergence of the optimization problem.

$$\begin{aligned} T_{\text{house}} &\geq T_{\text{house,set}} - \epsilon_{\text{house,LB}} \\ T_4 &\geq T_{\text{storage,set}} - \epsilon_{\text{storage}} \end{aligned} \tag{9}$$

Slightly different to the cost function used in the simulation model, equation 10 also includes w_{storage} and w_{house} as weighing vectors for the errors and ϵ_{i} which are chosen to get sufficiently small deviations from the calculated set points while affecting the cost function as little as possible.

$$\begin{aligned} costs_{total,MPC} &= P_{toGrid} \times costs_{consumption} - \\ P_{fromGrid} \times costs_{feedin} &+ \epsilon_{\text{house}} \cdot w_{\text{house}} + \epsilon_{\text{storage}} \cdot w_{\text{storage}} \end{aligned} \tag{10}$$

3.4. Data Basis

For the controller training and evaluation, artificially generated data that is based on measured real world data is used. It is generated for one year starting at the first of January and covers all input values needed for the simulation model: the variable energy tariffs for consumption and feed-in, the household load, the production of the hydroelectric power plant, the PV system production and voltage and the respective weather data. All methods used to calculate these values are explained below in more detail and plots from the data are shown in figure 4.

3.4.1. *PV production and Voltage data*

The generated PV production data is based on a MATLAB Simulink simulation model which is parametrized with the exact PV plant parameters from the building and recorded Meteonorm weather data from 2018 for ten different locations in the vicinity of the real world building. For the artificial PV production data, two of those weather datasets are randomly selected and averaged. If the new dataset length is up to one year, the averaged weather data is shortened to the desired dataset length. If it is longer than 365 days, the selection and averaging of the weather data is done until the desired length is reached. This artificially generated weather data file is then used as input for the PV simulation model, which then generates the desired PV production and voltage data.

3.4.2. *Hydroelectric Power Plant Production*

For generating the artificial hydroelectric power plant data, measured data from another power plant a few kilometres upstream of the original one is used, including the energy production and energy loss. This data was recorded from 2010 until 2015 and gets split up into single days of data, which are grouped by their month of measurement. By randomly selecting single days from the respective months that the new dataset should contain, subtracting the loss data from the production data and finally appending this days to one after another, the new dataset is built up.

3.4.3. *Household load*

The household load of the real building is approximated using the program LoadProfileGenerator (LPG) (Pflugradt, 2016), in which the building was modelled as precisely as possible so that the annual energy consumption is approximately the same. For that, also input data like the weather data and the production of the hydroelectric power plant were included. Running the LoadProfileGenerator then geneartes realistic load curves of the specified building for different start dates and lengths.

3.4.4. *Variable Energy Tariffs*

As basis for the artificially generated variable energy tariffs, aWATTar (aWATTar) data from 2015 until 2018 is used, including the energy consumption and feed-in tariffs. The data gets split up month-wise and collected in monthly pools, from which then randomly a monthly dataset is selected and appended to the previous ones until the desired length of the new dataset is reached.

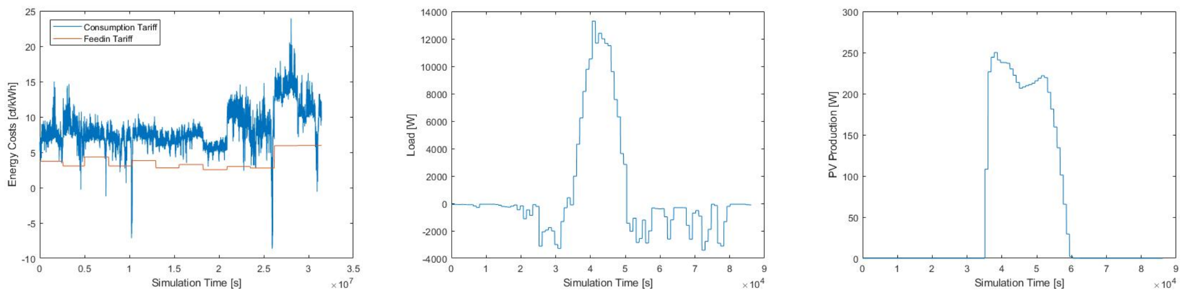


Figure 4. Visualizations of the artificially generated data basis for the energy tariffs for the whole year (left), the load including the hydroelectric power plant production for the 2nd of January (middle) and the PV production data for the 2nd of January (right).

3.5. Evaluation

In order to evaluate the heuristic energy flow controllers, the first day of the artificially generated data basis as explained in chapter 3.4 and two different genetic algorithms are used to each train five controllers, resulting in a total of ten heuristic controllers. The first algorithm is the single-objective Offspring Selection Genetic Algorithm (OSGA) from Affenzeller et al. (Affenzeller and Wagner, 2005), which just minimizes the energy costs of the building. As parameters, maximum 100 generations and 250 000 evaluated solutions, a mutation probability of 30%, a maximum selection pressure of 100, a population size of 500 with 1000 selected parents and a *GenderSpecific Selector* (Wagner, 2005) with a *ProportionalSelector* as female and a *RandomSelector* as male selector are used.

The second genetic algorithm that is used to train the heuristic controllers is an adaptation by Kommenda et al. (Kommenda et al., 2016) of the multi-objective Non-Dominated Sorting Genetic Algorithm (NSGA-II) originally developed by Deb et al. (Deb et al., 2002), which not only tries to minimize the energy costs of the system but at the same time also the complexity of the controllers, i.e. the symbolic regression trees contained in the solution candidates. (Kommenda et al., 2016) Similar to the parameters used for the OSGA, for the NSGA-II a maximum of 100 generations with a crossover probability of 100%, a mutation probability of 30% and a population of 500 solution candidates with 1000 selected parents is used. As selector, a *CrowdedTournamentSelector* (Deb, 2001) with a group size of six is used. Despite that, for the symbolic regression trees a maximum tree depth of 50 and a maximum tree length of 100 is specified, together with the following mathematical operators as grammar: the four arithmetic functions addition, subtraction, multiplication and division, the trigonometric functions sine, cosine and tangent, exponential and logarithm operators and the power functions square, power, square root and root.

For the evaluation of all energy flow optimizers, the described simulation model is used with a simulation interval of one second and an initial state of charge of the battery of 30%, which is also used for the training of the heuristic controllers. Each controller is simulated with this model for 30, 60, 180 and 364 days, starting at the day after the training on the 2nd of January. The result of the simulation, the energy costs of the system, is then used for the comparison of the three energy flow controllers: the heuristic controllers, the also previously described linear model predictive controller and the rule-based Fronius self consumption optimization (SCO).

4. Results and Discussion

As shown in table 1, the model predictive controller achieves better results than the Fronius self consumption optimization for all four evaluation timespans on average by 6.91% for 30, 60 and 180 days of simulation and by 0.51% for 364 days of simulation. Comparing the heuristic controllers to the SCO, it is shown that all of them also work highly significantly (average p=0.0000042) better for all evaluation timespans up to 180 days. When comparing them to the MPC, the heuristic controllers achieve significantly better results for 30 ($p = 0.00094$) and 180 ($p = 0.0032$) days of evaluation, while they achieve significantly worse results for 60 ($p = 0.59$) days of evaluation.

Taking a closer look on the heuristic controllers trained with the NSGA-II algorithm, it turns out that they perform worse than the ones trained with the OSGA algorithm. Comparing them to the SCO for 30, 60 and 180 evaluation days, the OSGA-trained heuristic controllers always achieve statistically significant better results with an average p-value of 0.00012, while the NSGA-II trained controllers there have an average p-value of 0.093. Comparing them to the MPC, this performance difference becomes even more obvious. The NSGA-II trained controllers achieve average energy costs of 1525.86€ (SD: 93.95€) for 30 days, while the OSGA trained controllers cause on average 1491.72€ (SD:23.68€) energy costs. This makes the OSGA controllers perform significantly better than the MPC with a p-value of 0.00035 while for the NSGA-II controllers no statistical significance could be proven

	Simulation Days			
Optimizer	**30**	**60**	**180**	**364**
SCO	1731.47	3382.71	6042.25	10248.33
MPC	1611.06	3147.38	5630.04	10195.95
NSGA-II 1	1471.49	3064.20	5239.43	11007.37
NSGA-II 2	1483.02	3142.76	5218.01	10801.48
NSGA-II 3	1489.11	3106.76	5328.83	11199.21
NSGA-II 4	1492.35	3109.56	5311.22	11212.22
NSGA-II 5	1693.32	3349.01	5939.66	10338.72
OSGA 1	1472.10	3063.97	5229.18	10999.01
OSGA 2	1526.10	3174.58	5419.97	11337.61
OSGA 3	1486.86	3103.76	5297.82	11132.25
OSGA 4	1469.41	3057.86	5216.16	10966.10
OSGA 5	1504.13	3149.4	5379.84	11281.45

Table 1. The energy costs in € for 30, 60, 180 and 364 days of simulation for the two reference optimization algorithms and the five controllers trained with each of the two genetic algorithms.

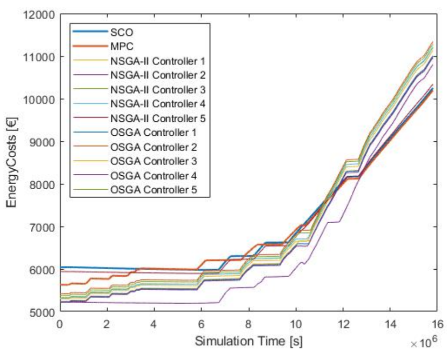
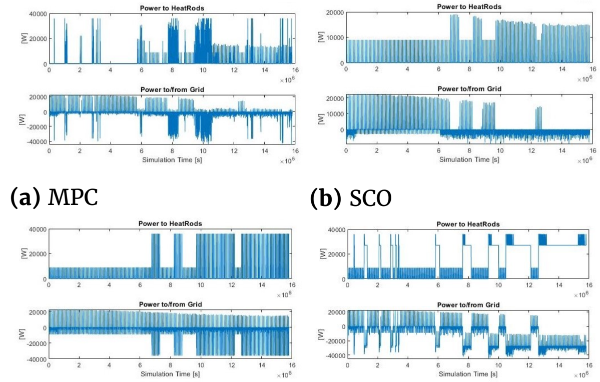


Figure 5. The energy costs for all evaluated energy flow controllers for the timespan between 180 and 364 evaluation days. The power consumed from the grid is here shown as negative values while the power fed into the grid is shown as the positive values.



(a) MPC **(b)** SCO

(c) NSGA-II controller #5. **(d)** OSGA controller #4.

Figure 6. The behaviour and influence of the different energy flow controllers on the heat rod power and the energy consumption from and feed-in to the grid.

with a p-value of 0.11. The same effect holds for 180 days of evaluation, where the NSGA-II trained controllers achieve average energy costs of 5407.43€ (SD: 301.17€), which are not statistically significant (p = 0.17) better than the MPC, while the OSGA trained controllers achieve significantly (p = 0.0013) better average energy costs of 5308.59€ (SD: 90.07€). For all evaluation timespans up to 180 days, the energy flow controller #4 trained with the OSGA achieves the overall best results.

However, for 364 days of evaluation, all heuristic controllers perform significantly worse than both reference optimization algorithms, which means that there is a turnover point somewhere between 180 and 364 simulation days. As shown in figure 5, this happens after approximately 11 000 000 seconds of simulation (calculated from 180 simulation days as starting point), which refers to approximately the beginning of November in the dataset. When analysing the behaviour of the different controllers in detail in order to find reasons for this turnover, it is found that their behaviour mainly differs in the usage of the heating rods. As shown in figure 6, the model predictive controller barely activates the heating rods at the beginning and mainly in the second half of the timespan while the SCO enables one of the heating rods with 9kW already in the first half. For these activations, the MPC uses power from the grid in addition to the one from the PV system, while the SCO supplies the heating rods mainly through the PV system while only consuming the energy needed to supply the household loads from the grid. In comparison to that are the heuristic controllers mainly using the grid to supply the activated heating rods and they also activate at least one of them right from the beginning of the shown timespan. This behaviour is most likely caused by the very short training time of just one day for the heuristic controllers, which is definitely not capable of covering all possible system states. Therefore, these controllers also slightly lack in their generalizability and ability to well handle evaluation timespans that differ from the training data.

5. Conclusions

This paper presents an approach for optimizing complex thermal electrically coupled buildings using genetic programming to train heuristic energy flow controllers with the goal to minimize its energy costs. For that, the building is modelled in MATLAB Simulink as precisely as possible and then used for the training process of the heuristic controllers. Using two different genetic algorithms, in total ten heuristic controllers are trained with artificially generated data of one day and evaluated for their energy cost optimization ability. As reference energy management systems, an existing rule-based self consumption optimization as well as a linear model predictive controller, which was also developed in the course of this work, are used and compared to the heuristic energy flow controllers. All energy management systems are evaluated in simulation with the same model which is also used for the heuristic controller training using artificially generated data for one year. The evaluated timespans start after the one day training data for the heuristic controllers on the 2nd of January and include 30, 60, 180 and 364 days of simulation

It is found that the linear MPC works better than the rule-based self consumption optimization for all evaluation timespans by saving up to 7.47% of the energy costs. The heuristic controllers work significantly better than both reference energy management systems for all evaluation timespans up to 180 days and can there save up to 8.79% of the costs compared to the linear MPC, while they perform significantly worse for 364 days and cause up to 10.08% more energy costs compared to the SCO. This turnover between 180 and 364 evaluation days is caused by the different usage of the heating rods, which are activated more often especially with the full 36 kW by the heuristic controllers and which are mainly supplied by energy consumed from the grid. In comparison to that, the MPC and the SCO use the heating rods less, do not activate all of them at once and try to supply them from the energy produced by the PV system as good as possible.

However, the very short training time of just one day of data needs to be taken into account here. Related works with a simpler system to be optimized have proved that longer training times for the heuristic controllers can bring big improvements. This is due to the bigger data variety included in longer datasets and because of that also a better generalization and optimization ability of the trained controllers. Therefore, in future work the heuristic controllers should be trained with additional datasets with different lengths so that this assumption can also be proved for this complex, thermal electrically coupled system. Despite that, a different data basis with measured instead of artificially generated data would be favourable to better approximate the behaviour of the real world building and to eliminate possible errors and wrong assumptions in the artificially generated data. Despite these limitations, it would also be interesting to test the approach with bigger simulations, for example including multiple buildings that should be optimized together. For that, more than three system parameters need to be optimized, which increases the complexity of the optimization problem remarkably. It is expected that the heuristic controllers then will perform better than the reference energy management systems, as they are much more flexible and do not need a linearisation of the problem like e.g. the model predictive controller does.

6. Funding

This project was financed by the European Regional Development Fund and the Province of Upper Austria. It was carried out by Fronius International GmbH and partners from the University of Applied Sciences Upper Austria.

References

(2020). *MATLAB version 9.3.0.713579 (R2020b)*. The Mathworks, Inc., Natick, Massachusetts.

Affenzeller, M. and Wagner, S. (2005). Offspring selection: A new self-adaptive selection scheme for genetic algorithms. In *Adaptive and natural computing algorithms*, pages 2187–221. Springer.

Alimohammadisagvand, B., Jokisalo, J., and Sirén, K. (2018). Comparison of four rule-based demand response control algorithms in an electrically and heat pump-heated residential building. *Applied Energy*, 209:167–179.

Arabali, A., Ghofrani, M., Etezadi-Amoli, M., Fadali, M. S., and Baghzouz, Y. (2012). Genetic-algorithm-based optimization approach for energy management. *IEEE Transactions on Power Delivery*, 28(1):162–170.

aWATTar. awattar. https://www.awattar.com/.

Chen, C., Wang, J., Heo, Y., and Kishore, S. (2013). Mpc-based appliance scheduling for residential building energy management controller. *IEEE Transactions on Smart Grid*, 4(3):1401–1410.

De Coninck, R., Baetens, R., Saelens, D., Woyte, A., and Helsen, L. (2014). Rule-based demand-side management of domestic hot water production with heat pumps in zero energy neighbourhoods. *Journal of Building Performance Simulation*, 7(4):271–288.

Deb (2001). Multi-objective optimisation using evolutionary algorithms. volume 16, page 247. John Wiley & Sons.

Deb, K., Pratap, A., Agarwal, S., and Meyarivan, T. (2002). A fast and elitist multiobjective genetic algorithm: Nsga-ii. *IEEE transactions on evolutionary computation*, 6(2):182–197.

Eseye, A. T., Zheng, D., Zhang, J., and Wei, D. (2016). Optimal energy management strategy for an isolated industrial microgrid using a modified particle swarm optimization. In *2016 IEEE international conference on power and renewable energy (ICPRE)*, pages 494–498. IEEE.

GmbH, F. I. Fronius energy management system. https://rb.gy/2fehaf. Accessed: 2020-08-27.

GmbH, F. I. Fronius ohmpilot. https://www.fronius.com/en/solar-energy/installers-partners/technical-data/all-products/solutions/fronius-solution-for-heat-generation/

fronius-ohmpilot/fronius-ohmpilot.

Godina, R., Rodrigues, E. M. G., Pouresmaeil, E., Matias, J. C. O., and Catalão, J. P. S. (2018). Model predictive control home energy management and optimization strategy with demand response. *Applied Sciences*, 8(3).

Gonçalves, I., Gomes, Á., and Antunes, C. H. (2018). Optimizing residential energy resources with an improved multi-objective genetic algorithm based on greedy mutations. In *Proceedings of the Genetic and Evolutionary Computation Conference*, pages 1246–1253.

Juelich, S.-I. (2018). Carnot toolbox ver. 6.2 2016b.

Kefer, K., Hanghofer, R., Kefer, P., Stöger, M., Affenzeller, M., Winkler, S., Wagner, S., and Hofer, B. (2019). A model-based learning approach for controlling the energy flows of a residential household using genetic programming to perform symbolic regression. In *International Conference on Computer Aided Systems Theory*, pages 405–412. Springer.

Kefer, K., Hanghofer, R., Kefer, P., Stöger, M., Hofer, B., Affenzeller, M., and Winkler, S. Simulation-based optimization of residential energy flows using genetic programming to solve a symbolic regression problem. in preparation for submission to the Journal of Energy and Buildings.

Kommenda, M., Kronberger, G., Affenzeller, M., Winkler, S. M., and Burlacu, B. (2016). Evolving simple symbolic regression models by multi-objective genetic programming. In *Genetic Programming Theory and Practice XIII*, pages 18–129. Springer.

Kothare, M. V., Balakrishnan, V., and Morari, M. (1996). Robust constrained model predictive control using linear matrix inequalities. *Automatica*, 32(10):1361–1379.

Morganti, G., Perdon, A., Conte, G., Scaradozzi, D., and Brintrup, A. (2009). Optimising home automation systems: A comparative study on tabu search and evolutionary algorithms. In *2009 17th Mediterranean Conference on Control and Automation*, pages 1044–1049. IEEE.

Pedrasa, M. A. A., Spooner, T. D., and MacGill, I. F. (2010). Coordinated scheduling of residential distributed energy resources to optimize smart home energy services. *IEEE Transactions on Smart Grid*, 1(2):134–143.

Pflugradt, N. (2016). *Modellierung von Wasser und Energieverbräuchen in Haushalten*. PhD thesis.

Salpakari, J. and Lund, P. (2016). Optimal and rule-based control strategies for energy flexibility in buildings with pv. *Applied Energy*, 161:425–436.

Seal, S., Boulet, B., and Dehkordi, V. R. (2020). Centralized model predictive control strategy for thermal comfort and residential energy management. *Energy*, 212:118456.

Sisodiya, S., Kumbhar, G. B., and Alam, M. N. (2018). A home energy management incorporating energy storage systems with utility under demand response using pso. In *2018 IEEMA Engineer Infinite Conference (eTechNxT)*, pages 1–6.

Srinivas, M. and Patnaik, L. M. (1994). Genetic algorithms: A survey. *computer*, 27(6):17–26.

Wagner, S. (2005). Sexualga: Gender-specific selection for genetic algorithms.

Wagner, S., Beham, A., Kronberger, G., Kommenda, M., Pitzer, E., Kofler, M., Vonolfen, S., Winkler, S., Dorfer, V., and Affenzeller, M. (2010). Heuristiclab 3.3: A unified approach to metaheuristic optimization. In *Actas del séptimo congreso español sobre Metaheurísticas, Algoritmos Evolutivos y Bioinspirados (MAEB'2010)*, page 8.