Carl von Ossietzky
Universität
Oldenburg

Deutsches Zentrum
für Luft- und Raumfahrt
Institut für
Vernetzte Energiesysteme

Master Engineering Physics

MASTER THESIS

Comparison of the Behavior Tree Concept with other Control Concepts used for Decentralized Energy Systems

Submitted by	Piet Urban Auguststraße 91 26121 Oldenburg Matriculation number: 6419517 piet.urban@uni-oldenburg.de
First examiner	Prof. Dr. Carsten Agert carsten.agert@dlr.de
Second examiner	Dr. Peter Klement peter.klement@dlr.de

Oldenburg, May 6, 2024

Contents

Abstract iii

Acronyms v

List of Figures vii

List of Tables ix

1 Introduction 1

2 Theory 5

2.1 Control Systems 5

2.1.1 Finite State Machines 7

2.1.2 Decision Trees 8

2.1.3 Behavior Trees 9

2.1.4 Model Predictive Control 12

2.2 Comparison of Concepts 13

2.2.1 Modularity Reactivity Trade-off 13

2.2.2 Expressiveness and Safety 15

2.2.3 Software Quality: Maintainability 16

2.3 Energy System 17

2.3.1 Technologie Overview 17

2.3.2 Control Requirements 20

3 Methodology and Experimental Setup **21**

3.1 Data Basis 21

3.2 Energy System Dimensioning and Component Modelling . . . 24

3.3 Code Basis 28

3.3.1 Python Packages 28

3.3.2 Code Structure and Processing 31

4 Results **35**
4.1 Comparison of the Development Effort 35
4.1.1 States vs. Actions 35
4.1.2 Heat System 37
4.1.3 Electricity System 43
4.1.4 Combination of Both Systems 46
4.2 Investigation of Operation 49
4.2.1 Optimisation 49
4.2.2 Considerations on the Computation Time 51
4.2.3 Comparison to DT and Classical Control 53
4.2.4 Results 55
4.3 Exemplary Application Cases 60
4.3.1 Blocking of the Heat Pump 60
4.3.2 Peak Shaving and Storage Availability in the Electricity System 62

5 Discussion: Potential of Behavior Trees in DES **67**

6 Summary and Outlook **71**

Bibliography **73**

Affirmation **83**

Abstract

Behavior trees are a proven concept in the creation of complex task-switching control for robotic systems and non-player characters in the computer games industry. Requirements such as flexibility, maintainability, reusability of functionalities or expandability also apply to the control of decentralised energy systems. Despite this, there is a noticeable research gap regarding the application of behavior trees in that sector, which motivates the explorations of their potential within the scope of this work.

Based on the definition and modelling of three exemplary concise energy systems, general behavior tree structures are created; implementation is carried out using the Python library py_trees. The advantage of modularity compared to the development of finite state machines known from robotics is verified for these systems. With a view to minimising annual operational performance indicators such as electricity price and emissions, we conclude that behavior trees have no fundamental advantages over other methods of decision making, such as decision trees, and that the comparison group of linear optimisations performed with the oemof-solph package can only be partially approximated. It is shown that the proposed basic structures allow specific relevant use cases, such as component blocking or peak reduction, to be realised.

Acronyms

AI artificial intelligence

BT behavior tree

BB blackboard

COP coefficient of performance

DES decentralized energy system

DT decision tree

ES energy system

FSM finite state machine

GED graph edit distance

HFSM hierarchical finite state machine

HP heat pump

KPI key performance indicator

MILP mixed integer linear programming

MPC model predictive control

PV photovoltaic

ROS robot operating system

TES thermal energy storage

List of Figures

2.1 Layered hierarchy of control concepts. 6
2.2 Example of a simple FSM. 7
2.3 Exemplary pseudo code for the Fallback and Sequence nodes. 10
2.4 Layered combination of BT and FSM control as a solution for the chattering problem. 11
2.5 FSM representation of an arbitrary BT. 14
2.6 Mapping of a single DT decision to an equivalent BT. 15
2.7 Basic structure of a HP. 18

3.1 Heat and electricity demands and load duration curves for the two years of available data 2017 and 2020. 22
3.2 Correlation of day ahead prices and specific CO_2 emissions for 2017 and 2020. 23
3.3 Overview of the three investigated energy systems. 24
3.4 Modelling of the HP based on the COP characteristics from TESPy for an exemplary ambient temperature of 7°C. 26
3.5 Principal code and import structure. Relevant excerpt of the code directory. 32
3.6 Processing scheme, different data and activation frequencies. . 33

4.1 Distinction between two FSM concepts for representation and control of an overall system state. 37
4.2 Control structure for the heating system, realised with a FSM. 38
4.3 Principal representation of a system state based BT control for the heating system. 40
4.4 Representation of the component based approach for a BT control of the heating system. 40
4.5 Illustration of the HP switching behavior with a single temperature threshold, controlled with a reactive BT vs. non-reactive FSM. 41

4.6 Exemplary operation of the FSM and BT with different execution frequencies. 42
4.7 Adding of data behavior and grid component to the main BT. 43
4.8 FSM control structure for the electricity system. 44
4.9 Comparison of non-reactive FSM and BT operation with activations $\Delta t = 1000\,\mathrm{s}$. 45
4.10 System state based BT realisation of a control structure for the electricity system. 46
4.11 Proposed strategies for combination of the simple system control BTs for control of the full system. 47
4.12 Component based vs. system state based approach for combining the FSMs of single systems. 47
4.13 All optimisation results for 2020 and comparison of the load duration curves of single components. 50
4.14 Comparison of computational effort of the optimisation with oemof-solph including different HP models. 51
4.15 Development of BT execution time for different sets of subsequent simulation runs. 52
4.16 Component based approach to hysteresis switching of the HP. 53
4.17 Decision tree implementation using the BT framework. 54
4.18 DT training on 2017 heat system optimisations. 55
4.19 Comparison of the ten runs with lowest CO_2 emissions for each case and optimisation baseline. 56
4.20 Priority based control (BT) and introduction of a DT regressor for charging in the electricity system. 58
4.21 Comparison of exemplary BT runs with the optimisation for the full system. 59
4.22 BT for the heating system with safety tree for blocking. 60
4.23 Operation of the heat system BT with blocking for two exemplary days. 61
4.24 Sub tree of 4.10 with a modified battery condition. 62
4.25 Exemplary day showing the operation of a BT with peak shaving mechanism. 63
4.26 Effect of the peak shaving mechanism, shown in figure 4.24 for one year of operation. 63
4.27 Sub tree of 4.10 with a modified excess and battery condition. 64
4.28 Two sample days illustrating the operation of the battery charging mechanism, shown in figure 4.27. 64

List of Tables

2.1 Node types that a BT can be build of with conditions for their respective feedback. 9

2.2 Expected advantages and disadvantages of investigated control structures. 13

3.1 Overview of utilized weather data: ground temperature, direct and diffuse irradiation. 22

4.1 Synthesis of the heating system into single heat flows Q_k and their classification in terms of boundary conditions. 38

4.2 Synthesis of the electricity system into power flows P_k and their classification in terms of boundary conditions. 43

4.3 Costs applied to the optimisation problem for the different objectives. 49

4.4 Estimation of additional costs for two operation strategies of the HP. 57

4.5 Overall results for the three different energy systems with relative differences to the optimisation results. 59

1 Introduction

Transition to a Decentralized Energy System

The international decline in greenhouse gas emissions largely relies on the growth of renewable energies in the electricity and heating industries. With a share of 20.4% of gross final energy consumption, almost 232 million t CO_2 equivalent could be saved in 2022 in Germany compared to fossil energy generation [1]. Amongst others, the aim of the federal government is to increase the proportion of the electricity market to over 80% by 2030 [2, 3]. Due to locally available resources and the convergence of producers and consumers, this development is accompanied by a shift from centralised to highly decentralised energy supply [4].

This offers various benefits such as improved energy quality and reliability, reduced distribution needs, lower losses, and associated economic advantages. The overarching goals of energy security, affordability, and sustainability remain consistent [5]. From this viewpoint, decentralization and digitalization pose new challenges, notably the rise in actors and the shift from consumers to prosumers, leading to more complex systems [6]. Furthermore, the inherent intermittency and unpredictability of renewable energies present fundamental issues. In summary, there is a significant demand for flexibility in technologies and their control options, emphasizing the importance of system interpretability, along with considerations for maintainability and expandability [5, 6].

Behavior Trees

Proposing a potential solution to address above requirements of future system and component control, the concept of behavior trees (BT) stands out. BT enable a structure for switching between different tasks in an autonomous agents. They are characterized by high modularity, particularly in the reusable nature of individual subtrees or behaviors, and clarity, along with a high reactivity to changing environmental conditions. In this context, behavior trees established themselves since the mid-2000s in the computer game industry for non-player character (NPC) control. Pioneered by the works of Mateas [7], Isla and Champard [8], the concept has seen numerous extensions for creating artificial intelligence (AI) behavior [8, 9]. Especially the possibility to easily extend behavioral patterns offers a great advantage over the otherwise widespread control with finite state machines [9, 10].

With an increasing number of scientific publications outside industry, further application areas have been opened up for the use of BTs. Example fields, beyond the gaming realm, are humanoid robotics [11], automation of production [12] and support in brain surgery [13].
Only recently, their utility has also been proposed for application in the energy sector, more specifically in the control of smart grids [6] and the operation of microgrids [14]. Again, the focus is on ensuring stable and economic operation under a wide range of environmental and working conditions. As the demand for control solutions for decentralized energy system (DES) continues to grow, the theoretically well-researched and in diverse fields practically proven concept of behavior trees emerges as a promising candidate.

Content of this thesis

The agreement of the described requirements in the energy sector and properties of behavior trees is the motivation for an in-depth investigation and exploration of their potential in DES within the scope of this work. Following research questions were formulated:

- How can the concept of behavior trees be applied to decentralized energy systems and how can its advantages be leveraged?
- Can control with a behavior tree approximate model-based optimization in simple systems?
- What advantages can be identified compared to other control systems; for which operating conditions and special cases is the use of behavior trees particularly suitable with regard to stable system behavior?

Here, the focus of this work was clearly on control options for energy system components, not on further evaluation of technology selection and dimensioning.

First a number of concepts, i.e. finite state machine (FSM), decision tree (DT), BT and model predictive control (MPC) will be introduced within the theory chapter. Furthermore, important aspects of the relevant energy system technologies should be discussed here. The subsequent methodology chapter will be concerned with the applied data and component models, but also with the code and simulation environment developed during this thesis. Results of the investigation will be presented in chapter 4, whereby a distinction is made between comparison of development effort and comparison of operation of different control structures. Finally, the potential of behavior trees in decentralised energy systems will be discussed along with an outlook for further research.

2 Theory

A number of control concepts will first be described in this chapter. Some work has already been conducted on the comparison of such structures in other domains, mainly for BT and FSM, following is an overview of this. Thereafter a theoretical introduction to the investigated energy technologies will be given. Consideration of explicit modelling of the energy system (ES) is carried out in the third chapter.

2.1 Control Systems

The study of control systems is a conceivably large field including, in the broadest sense, any physical object. Leaving that level of abstraction, control systems may be identified as systems that *"dynamically or actively command, direct or regulate"* a number of outputs with respect to their inputs [15]. Nowadays, and here too, this often concerns digital systems of a discreet nature [16]. It is therefore worthwhile to first contextualise the concepts considered below. A hierarchical structuring, proposed by Marzinotto et al. [11] is shown in figure 2.1.

Here a subdivision into three layers is provided. Automated creation and updating of overarching plans takes place in the top layer. This can involve predictions, system optimisation or training, possibly including MPC and DT concepts of the subsequent sections. The middle layer represents the transmission to low-level hardware controllers (e.g. PID, two-point controller or fuzzy logic) in the bottom layer and essentially realises the switching between them; BT and FSM can be assigned to it [11]. Here, interaction of continuous variable dynamics, described by physical laws, and event-driven discrete variable dynamics, whose evolution depends on "if-then-else" rules, leads to complex dynamic behaviors [17].
It is nevertheless pointed out, that there is no clear demarcation between those layers, so that they can merge into one another in the application [11].

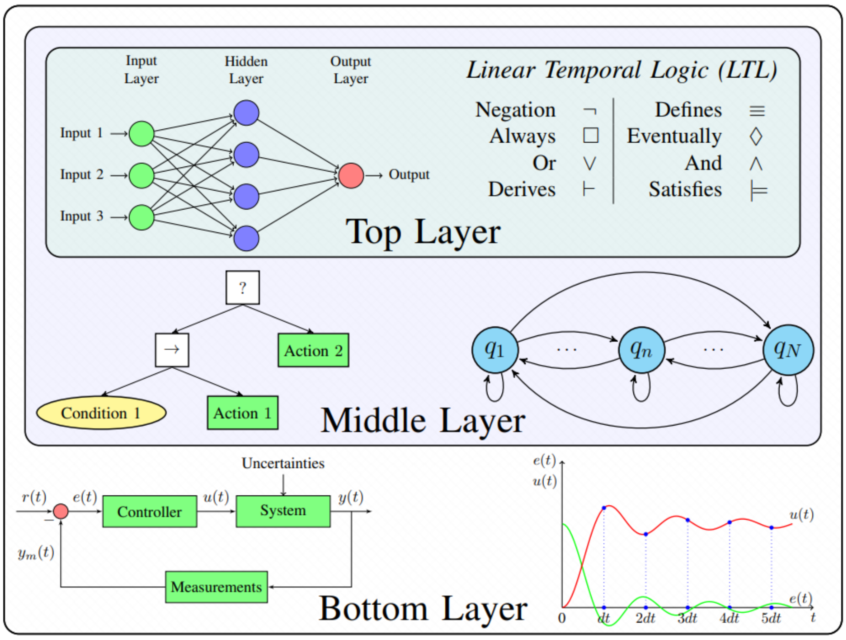


Figure 2.1: Layered hierarchy of control concepts. From high level decision making and plan execution (Top Layer) to low level controllers (Bottom Layer). Taken from [11].

Notations from Graph Theory

Various terms from graph theory are used to formulate the following concepts, of which a selection will be briefly presented. Further nomenclature can be found in the works of Priese and Erk [18] as well as Zelazo et al. [19].

- A *graph* $G = (V, E)$ consists of a set of *vertices* (or *nodes*) V and a set of *edges* $E \subseteq V \times V$. If for every nodes $a, b \in V : (a, b) \in E \Rightarrow (b, a) \in E$ holds, it is called *undirected*, otherwise G is *directed*.
- In connections $(v, v') \in E$ or $v \to v'$, v is called the *parent* of v' and v' the *child* of v. A *path* $v_1 \ldots v_n$ with $n > 1$ is called a *cycle* if $v_1 = v_n$.
- A directed acyclic graph $T = (V, E, v_0)$ with a designated root node is called a *tree*, if every node $v \in V$ of the graph can be reached from v_0 and no node has more then one parent.
- Further designations that are used in the context of trees are *inner nodes* having at least one child, *leaves* having no children and *branches*, which are the paths from inner nodes to leaves. The *depth of a tree* denotes the maximum length of a path within the tree and *sub trees* are defined as trees $T' = (V', E', v_0')$ with $V' \subseteq V$ and $E' = E \cap (V' \times V')$ [18, 19].

2.1.1 Finite State Machines

The concept of automata originates from theoretical informatics and the fundamental problem of recognition, i.e. analysis of formal languages. As the name suggests, a FSM is a very basic automata that consists of a finite number of states [18]. When reading an input (word), it is in exactly one of these states, depending on the last instruction (the last letter read). A FSM can accept or reject words based on the attainment of defined final states [18]. In a formal definition the FSM is a Tupel $A = \{K, \sum, \delta, s_0, F\}$ with:

- K the finite amount of states,
- $\sum$ the alphabet of the input,
- $\delta = K \times \sum \rightarrow K$ the transfer function,
- $s_0 \in K$ the initial state and
- $F \subseteq K$ the set of final states [18].

An exemplary FSM is shown in figure 2.2. Here applies $\delta(s_0, a) = s_1$, $\delta(s_1, b) = s_1$ and so on. Only s_0 is defined as a final state, therefore every input word containing an even number of a's would be accepted by this machine.

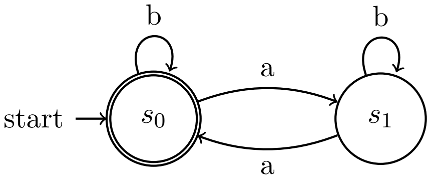


Figure 2.2: Example of a simple FSM. Taken from [18].

In an application and in the way it is used in the remainder of this thesis, event-based transitions between states primarily take place in consideration of the system state. Respective conditions correspond to entries from the alphabet $\sum$. Certain computations, which are usually associated with actions of the controlled agent, are executed in between. Depending on the implementation, a distinction can be made between execution upon entry into or exit out of a state [20], as shown in chapter 3.3.

There are several architectures based on FSM, one advancement that should be mentioned here is the hierarchical finite state machine (HFSM). HFSM aim at increasing modularity by introducing so-called super states, each of which summarises several system states. However, such subdivisions need to be user-defined and can be rather arbitrary [9].

2.1.2 Decision Trees

A DT is a tree in the sense of the above definitions, with each inner node having exactly two children. Each of these nodes is linked to a boolean function; the two children $\mathscr{D}_{i1}$ and $\mathscr{D}_{i2}$ of a parent $\mathscr{D}_i$, can themselves be sub trees or leaves (e.g. atomic actions or decision outputs) and are accessed with respect to the predicate C_i [18, 21].

$$\mathscr{D}_i = \begin{cases} \mathscr{D}_{i1} & \text{if predicate } C_i \text{ is true} \\ \mathscr{D}_{i2} & \text{if predicate } C_i \text{ is false} \end{cases} \tag{2.1}$$

As a consequence, exactly one leaf node is reached with each execution. Furthermore there is no information back flow from child nodes to their parents, which is seen as the main disadvantage by Colledanchise and Ögren [9].
Due to the unitary formulation DTs are a straightforward possibility for decision making, i.e. data classification. The comparison to BT appears natural in that both share the advantage of a modular hierarchical structure [9]. Details will be provide in the next section 2.2.

Decision Tree Learning and Regression

There exist a number of algorithms for training of DTs as concluded by Somvanshi et al. [22]. The focus is often on classification, i.e. categorisation into a finite set of groups based on certain characteristics. One exception that is also able to create a DT regressor for continuous data is the so called CART (Classification and Regression Trees) algorithm introduced by Breiman [23]. It should be described briefly.
Starting point is a number of training vectors $\mathbf{x}_i \in \mathbb{R}^n, i = 1 \ldots l$ with a corresponding label vector $\mathbf{y} \in \mathbb{R}^l$. By splitting a set of data Q_m with n_m samples at node m of a DT two subsets $Q_m^{left}(\Theta)$ and $Q_m^{right}(\Theta)$ can be obtained based on the boolean logic, equation 2.1. They can be described as follows given the candidate split tuple $\Theta = (j, t_m)$ with the threshold t_m and feature j [24].

$$Q_m^{left}(\Theta) = \{(x, y) | x_j \leq t_m\},\ Q_m^{right}(\Theta) = Q_m \setminus Q_m^{left}(\Theta) \tag{2.2}$$

Objective of the algorithm is the minimization of the function $G(Q_m, \Theta)$, i.e. finding the threshold and feature for lowest impurity of the respective subsets.

$$
\begin{aligned}
G(Q_m, \Theta) &= \frac{1}{n_m} \left[(n_m^{left} H(Q_m^{left}(\Theta)) + n_m^{right} H(Q_m^{right}(\Theta)) \right] && (2.3) \\
\Theta^* &= \arg\min_{\Theta} G(Q_m, \Theta) && (2.4)
\end{aligned}
$$

A simple loss function $H(Q_m)$ to calculate this impurity for continuous y is the mean squared error, equation 2.5. The procedure is repeated for each node and sub tree until a defined maximum depth is reached or until a subset remains empty [23, 24].

$$
H(Q_m) = \frac{1}{n_m} \sum_{y \in Q_m} (y - \bar{y}_m)^2 \tag{2.5}
$$

2.1.3 Behavior Trees

This work's central subject, the behavior tree has been introduced in chapter 1. It is strongly related to the two concepts already presented in this section. The objective of structuring and switching tasks of an agent corresponds to that of a FSM, structural similarity to DT will be obvious.

Table 2.1: Node types that a BT can be build of with conditions for their respective feedback. Control flow nodes process the feedback of their children. Based on [9].

Node type		**Succeeds**	**Fails**	**Running**
Control flow		*feedback from children*		
Sequence	→	All succeed	One fails	One running
Fallback	?	One succeeds	All fail	One running
Parallel	⇉	$\geq M$ succeed	$> N - M$ fail	else
Decorator	◇	Custom	Custom	Custom
Execution				
Action	▭	Upon completion	If impossible	During compl.
Condition	⬭	If true	If false	-

Just like DT, a BT can initially be defined as a directed, rooted and acyclic graph, i.e. as a tree (see sec. 2.1). Now, however, the number of children connected to an inner node is not restricted. Instead there exists a fixed order to each set of them. For reasons of clarity, the organisation in graphic representations often corresponds to execution from left to right and top (root of the tree) to bottom [8, 9, 25, 26].

In the standard formulation BTs consist of two types of nodes, control flow (inner) nodes and execution nodes, which will be the leaves of the tree [9]. They are in turn divided into a total of six types, presented in table 2.1.

As the tree is executed from its root and explored through a trigger often referred to as *tick*, every node will give a feedback *Running*, *Success* or *Failure* depending on its respective rule set [9]. A consequence is that in most cases not all sub trees or leaf nodes are reached as described subsequently.
It can be seen that *Fallback* and *Sequence* nodes correspond to logical `OR` and `AND` operations. The shown logic directly affects the execution of tasks in a BT. A sub tree with a Fallback root node is left with Success as soon as one child is successful, a Sequence on the other hand returns Failure or Running as soon as one child responds the same. This can be illustrated by the pseudo code in figure 2.3.

Algorithm 1 Sequence node with N children

```
for i ←1 to $N$ do
    $state$ ←Tick($child$(i))
    if $state = Running$ then
        return $Running$
    else if $state = Failure$ then
        return $Failure$
    end if
end for
return $Success$
```

Algorithm 2 Fallback node with N children

```
for i ←1 to $N$ do
    $state$ ←Tick($child$(i))
    if $state = Running$ then
        return $Running$
    else if $state = Success$ then
        return $Success$
    end if
end for
return $Failure$
```

Figure 2.3: Exemplary pseudo code for the Fallback and Sequence node types. Pseudocode for all node types can be found in [11, 27].

BTs can also be defined in a more formal Statespace formulation as a Tupel $\mathscr{B}_i = \{f_i, r_i, \Delta t\}$ with:

- $f_i : \mathbb{R}^n \rightarrow \mathbb{R}^n$ an ordinary difference equation,

$$x_{k+1} = f_i(x_k) \tag{2.6}$$

$$t_{k+1} = t_k + \Delta t \tag{2.7}$$

- $r_i : \mathbb{R}^n \rightarrow \{\mathscr{R}, \mathscr{S}, \mathscr{F}\}$ the return function, mapping to one of the states Running $\mathscr{R}$, Success $\mathscr{S}$ or Failure $\mathscr{F}$ and
- Δt the time step [9].

Here the system state $x_k = x(t_k)$ and different regions R_i, S_i, F_i of a system are defined with respect to the response [9, 28]:

$$R_i = \{x : r_i(x) = \mathscr{R}\},\ S_i = \{x : r_i(x) = \mathscr{S}\},\ F_i = \{x : r_i(x) = \mathscr{F}\} \tag{2.8}$$

One node type that is not further specified in table 2.1 is the Decorator, which modifies the behavior of its single child. Examples are the multiple execution, the inversion of the response or the delay of certain actions to name only a few. The Decorator provides an outlook on how flexibly and extensively the concept can be further developed. With it, the possibilities of a "classical" tree are considerably expanded.

Extensions and Hybrid Approaches

Multiple further extensions for BT are proposed, e.g. k-BTs in [29], stochastic BTs [9] or more flexible trees [8]. Another suggestion is the use of nodes with memory for certain applications. This is connected to the major downside that reactiveness can cause in a pure BT, the possibility of rapid fluctuation between tasks, e.g. primary job and recharging in a robot operating system (ROS), see. figure 2.4. Yet a further conceivable solution to such problems is the strength-orientated combination of BT and FSM [29].

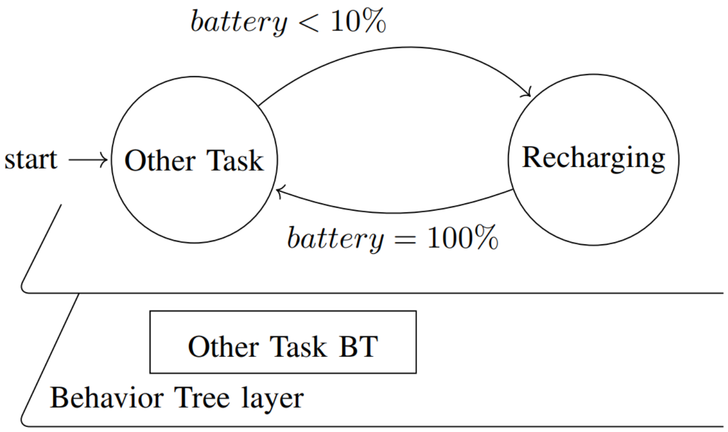


Figure 2.4: Layered combination of BT and FSM control as a solution for disadvantageous reactiveness (chattering problem). Taken from [30].

2.1.4 Model Predictive Control

A last method that should be introduced here is that of MPC. It can rather be assigned to the top layer in the picture 2.1, as generation and update of an overall (future) plan for the operation is inherent [11]. Nowadays, there are a large number of multiple input, multiple output systems for whose control design MPC can be usefully applied. Amongst others, this is due to the fact, that large requirements of real time computing can be handled increasingly well [31].

The basic idea consists of taking certain future predictions into account. As with other control concepts, this requires knowledge of the controlled processes, beyond that a sometimes complex optimization process. It is central to MPC to set a certain objective or cost function ℓ, the below sum J_N is to be minimized with respect to constrained set of control variables u within the time horizon of N discrete steps. The optimized control input will affect the system state of later timesteps as $x(k+1)$ after known initial state $x(0) = x_0$ [31].

$$J_N(x_0, u) = \sum_{k=0}^{N-1} \ell(x(k), u(k)) \tag{2.9}$$

$$x(k+1) = f(x(k), u(k)) \tag{2.10}$$

Equation 2.9 illustrates a major challenge for MPC applications. All objectives of a control system need to be embedded in the cost function $\ell(x_k, u_k)$. This is certainly viable if the parameter to be minimised is obvious or singular, e.g. the total price or emissions. When it comes to the extensibility of more specific functionalities, however, the definition of ℓ and corresponding implementation might be difficult. Especially for discrete decision making, computational effort can be quite high [31].

Conclusion

As the focus of this work is on comparison of the presented concepts, expectations of the main advantages and disadvantages should be briefly summarised here. A more detailed discussion of some aspects will be conducted in the next section. An overview is already provided in table 2.2.

	FSM	**DT**	**BT**	**MPC**
Scalability, Reusability	−	+/−	+	+/−
Interpretability	+	+	+/−	−
Safety, Failure Handling	+/−	−	+	−
Computational Effort	+	+	+	−
Optimal Operation	−	−	−	+

Table 2.2: Expected advantages and disadvantages of investigated control structures.

BT promise to have favourable properties due to their high modularity. All of the first three structures in table 2.2 can be considered interpretable, although DT and FSM may appear more intuitive initially. In the other aspects shown, FSMs are more likely to be disadvantageous. Particularly in terms of scaling and reusability, but also with regard to error handling, this can be explained by the expected high number of additional connections required when modifiying a respective control structure. In the latter aspect, DT are also viewed critically due to the lack of information feedback from the leaf nodes. Implementation of more complex systems is questionable here.

In the case of MPC, as already mentioned, difficulties arise with such criteria, predominantly in the formulation of a suitable cost function. The reason for the decision in favour of a particular operational output is often incomprehensible. Its main advantage of optimal operation (in terms of the defined loss function) is connected to potentially high computational effort as will be investigated.

2.2 Comparison of Concepts

A lot of research on comparing BT, DT and FSM has already been conducted, both in the context of their application as well as on a theoretical basis. Here, some relevant aspects from the above conclusion should be discussed qualitatively to a slightly broader extent. The introduction of quantitative measures, on the other hand, is rather difficult.

2.2.1 Modularity Reactivity Trade-off

Reactivity and Modularity are considered the two main principles and advantages of BT [30]. The former refers to the ability of a controlled agent to react very attentively and fastly to its environment, i.e. to changes in environmental

conditions. Modularity concerns the development of a structure and the possibility of assembling or modification of individual parts without great effort. These aspects become apparent in comparison with FSM. FSM are based on so called one-way control transfers or `GOTO` statements [18]. Here modularity is reduced when attempting to increase reactivity, as this inevitably requires the addition of such transitions. BTs, on the other hand, are said to follow the trend of modern programming languages in utilizing two-way-control-transfers [9, 32].
A central objective in this area of conflict is the reduction of programming effort [33, 34]. Although there is always the possibility to translate a BT into a FSM representation (as seen exemplary in figure 2.5) and vice versa, it is not necessarily sensible.

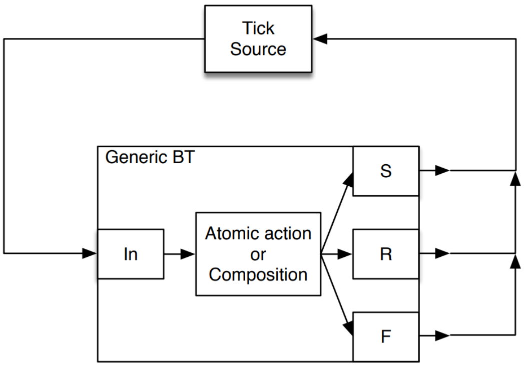


Figure 2.5: FSM representation of an arbitrary BT. Reverse translation of FSM into BT formulation can be found in [9]. Taken from [9].

Concrete formulation and formal comparison of modularity of DTs, BTs and further control architectures is investigated in depth in the work of Biggar et al. [29] by introducing the generalized term of decision structures. Reference to the programming effort is made in the publication of Iovino et al. [33]. Two measures that are utilized there will be briefly presented.

Complexity Measures from Software and Graph Theory

The cyclomatic complexity CC was initially proposed as a software metric by McCabe [35]. Here, the number of edges E, nodes N and subsystems (separate graphs) L are utilised to give a simple measure of interdependency or linkage within a structure. Trees always have a CC of 1 [29, 36].

$$CC = E - N + 2L \tag{2.11}$$

A possibility for comparing two graphs G_1 and G_2 is the determination of the so called graph edit distance (GED). It is defined as the minimum costs c for editing, removing or adding of nodes and edges (set γ) in order to bring them equal [37].

$$GED(G_1, G_2) = \min_{e_1,\dots,e_k \in \gamma(G_1,G_2)} \sum_{i=1}^{k} c(e_i) \tag{2.12}$$

The following two sections conclude further consequences of the above discussed aspects reactivity and modularity. They are fairly fundamental as noted by Biggar et al. [30].

2.2.2 Expressiveness and Safety

Expressiveness can be defined as the possibility to construct certain behavior with a given set of actions, as formulated in more detail by Biggar et al. [38]. In the provided framework it was proven there that pure BT are as expressive as pure DT, but less expressive than FSM [38]. This argument can be depicted, at least for the tree structures within figure 2.6. In order to create the same behavior, action nodes of the BT are, however, restricted to the Running feedback, emphasising the major drawback of missing information back flow in DTs [9].

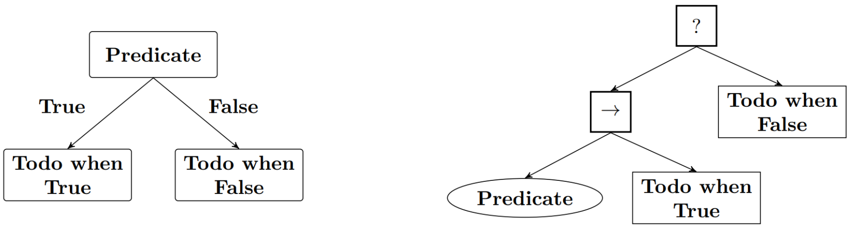


Figure 2.6: Mapping of a single DT decision (left) to an equivalent BT (right). This requires actions always to return Running. Taken from [9].

The lower expressiveness compared to FSM can be explained by a lack of memory in BT, i.e. missing information “stored” in the current state of the first. BTs are to a certain extent relatively rigid in terms of prioritising tasks. One way of preventing the constant re-execution of certain nodes (i.e. including tasks that have already been completed) is to introduce memory nodes. Feedback from previous activations is saved here, which corresponds to the FSM property described above. However, as recommended by Colledanchise and

Ögren [9], they should not be deployed as soon as the task switching occurs in an unpredictable environment where the possibility of a quick redirection to previous subtrees is desired.

Another important aspect in this context is safety. Safety can be defined as the ability of an autonomous agent to avoid certain regions of the state space (which is admittedly more descriptive in robotic applications) [25]. Reactivity is extremely advantageous for safety concerns, as undesired actions can be quickly interrupted or omitted [9, 25]. A verification method for safety properties of BT is proposed by Henn et al. [26].

2.2.3 Software Quality: Maintainability, Reusability, Expandability, Interpretability

A high modularity of a control structure entails advantageous aspects, Maintainability, Reusability, Expandability and as already noted Interpretability, as described in the introduction. Whilst there exist an intuitive understanding of such terms, finding a tangible definition is rather problematic. The ISO/IEC 25010 standard defines software quality characteristics, amongst them the above selection within the category "Maintainability" [39]. Criticism and comment of difficulties in application are already indicated by Estdale and Georgiadou [39], both for the older and the revised standard.

A comparison with introduction of a score system based on ISO/IEC 25010 as in the work of Dang [40] might be applicable for the development evaluation, i.e. quality management of complete software products. For the comparison of individual characteristics, this approach is likely to become too simplistic, subjective or even manipulable. Other specific measures as e.g. the so called Maintainability index introduced by Coleman et al. [41] and applied a.o. by Olsson [34] for the comparison of BT and FSM emphasise above described problems. The rapid advancement of tools and usability of existing comprehensive libraries devalues metrics such as *lines of code* or *number of comments per module* [33].

2.3 Energy System

It has been seen that BTs are mainly investigated in the context of games and robot applications and that the transfer to the energy sector is just emerging. The scope of this section is to briefly introduce the technologies being used in the simulation later in this work. Actual modeling of the components will be described in the next chapter.

2.3.1 Technologie Overview

PV Systems

Utilization of solar energy can be achieved by solar thermal and photovoltaic systems. Characteristic for the latter is the direct transformation of solar to electrical energy (build up of an electrical potential) without intermediate processes or rotating masses [42].
The actual energy output is limited by the main factors of radiation and efficiency of the photovoltaic (PV) cell used. Irradiation is divided into direct and diffuse shares and subject to strong fluctuations, based on seasonal and daily periodicity, cloud formation, shading and surface orientation. Corresponding to the typical current-voltage curve of the cell (see Shockley's diode equation), the maximum power point or peak power P_{peak} depends approximately linearly on the irradiation [43, 44]. Efficiencies in practical application range between 10% and 19%, solar heating of the cells, reflection losses and other effects can reduce these values (note the standard testing conditions $1000\,\mathrm{W\,m^{-2}}$ and cell temperature of 25°C) [43, 44].

Heat Pump

In a heat pump (HP), based on the left-hand Carnot process, mechanical work is performed to further cool a lower temperature and heat a higher temperature reservoir. This allows low-temperature heat to be utilised, e.g. from near-surface, waste water or ambient air. The coefficient of performance (COP) ϵ_{h}, representing the ration between usable heat $\dot{Q}$ and invested work is the decisive factor for reasonable application [42, 43].
A schematic representation of the principal components of a heat pump is shown in figure 2.7.

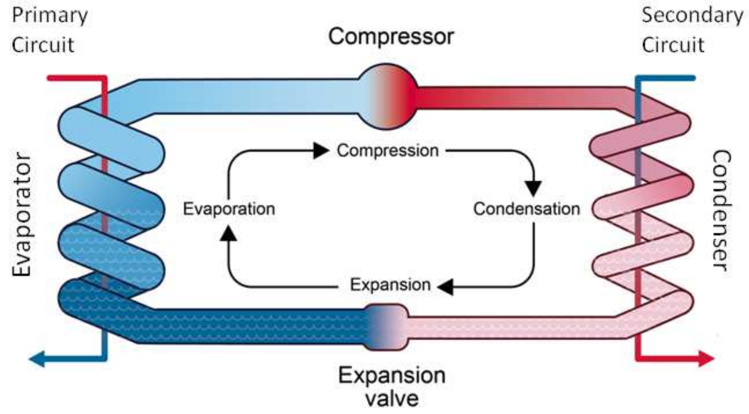


Figure 2.7: Schematic representation of the basic structure of a HP. The thermodynamic cycles consists of four main phases. Taken from [45].

Isentropic compression in the gas phase within the compressor is followed by an isobaric liquification of the working fluid (or refrigerant) in the condenser. Here the useable heat is rejected. Relaxation in the expansion valve and vaporisation at lower temperature complete the cycle [42, 46]. As mechanical work is performed on the refrigerant only at the compressor (P_{comp}), ϵ_{h} can be calculated as follows. It is fundamentally limited by the reciprocal of the Carnot efficiency ϵ_{max}, based on the temperature levels of the respective reservoirs T_{high} and T_{low} [42].

$$\epsilon_{\mathrm{h}} = \frac{\dot{Q}}{P_{\mathrm{comp}}} < \epsilon_{\mathrm{max}} = \frac{T_{\mathrm{high}}}{T_{\mathrm{high}} - T_{\mathrm{low}}} \tag{2.13}$$

Batteries and Accumulators

Deployment of storage in an ES makes it possible to use energy flexibly both in terms of space and time, whereby the latter is likely to be decisive when it comes to setting up an operating strategy.

Electrochemical energy storage is based on chemical reactions in which electrical charges are transferred. They are typically constructed from electrodes that are connected to each other via an electrolyte as an ion-conducting phase. Units that can be charged and discharged are referred to as secondary batteries or accumulator [47].

Important characteristics of a battery are the nominal voltage, which depends on the deployed electrodes and the energy content. The latter is often specified as a charge quantity (A h), since the extractable capacity (W h) depends

on the discharge conditions. An important practical aspect is the charge resp. discharge power. It is also dependent on the battery status and often restricted by the manufacturer [47]. A measure for this is the so called C-rate, which is defined as the ratio between charge current and energy content [43].
Cyclic ageing of a battery depends on the applied C-rates, temperature and depth of discharge. For the widely used Lithium-ion battery lifetimes ($\approx 80\%$ useability of initial capacity) of typically 500-800 [43] and up to 10^4 cycles [48, 49] are achieved. In measurements [47] depths of discharge between 30% and 70% led to the best performance in this context. The self-discharge rate for Li-ion batteries can be very low ($<3\%\mathrm{a}^{-1}$) at moderate temperatures and medium state of charge [47].

Thermal Energy Storage

Given the share of the heating sector in final energy consumption, thermal energy storage (TES) systems play a decisive role in renewable ES. A very common storage medium in sensible storages without phase transition is water, given its high specific heat capacity c_P, low costs and sustainability. Typical temperatures in the building sector range from 25°C for floor heating, to 60°C for for water supply and up to 90°C in some radiators [47]. The storable amount of heat Q for a specific temperature range $T_\mathrm{high} - T_\mathrm{low}$ is given by

$$Q(T_\mathrm{s}) := E_\mathrm{th}(T_\mathrm{high} - T_\mathrm{low}) = \rho c_\mathrm{P} V (T_\mathrm{high} - T_\mathrm{low}) \tag{2.14}$$

depending on density ρ and volume V of the deployed medium [47, 50].
Thermal insulation plays an important role in sensitive TES. According to the textbook by Goeke [51], 65-80% of heat losses are caused by heat conduction within the outer surfaces, another important share is made up of connection losses. The heat flow $\dot{Q}_\mathrm{loss}$ can be derived by the conducting area A, temperatures of storage and environment T_s and T_env as well as the heat transition coefficient u [47, 51].

$$\dot{Q}_\mathrm{loss} = uA(T_\mathrm{s} - T_\mathrm{env}) \tag{2.15}$$

2.3.2 Control Requirements

Some requirements for control of these technologies arise quite naturally from the above descriptions. Returning to the segmentation shown in figure 2.1, the main purpose of the middle layer control structures is setting of power levels or switching components on and off [14]. A variety of control opportunities concerning e.g. the mentioned MPP tracking in a PV converter [52] is more likely to be assigned to the lower layer, note that there are very diverse approaches at this level as concluded for example by Murillo-Yarce et al. [52]. Some requirements should be listed conclusively:

- In heat pumps power control within a certain range is possible for example with a so called scroll compressor [43]. They are not normally in continuous operation, which is why special attention must be paid to the switch-on and switch-off processes. A minimum on or off time of e.g. 15 min is often required to reduce wear [53].
- In addition to the power output, monitoring the status or temperature is important for the storage systems presented. Particularly in the case of Li-ion batteries, deep discharge can damage the electrode. A battery management system would possibly also include the monitoring of cell voltage and temperature. It could be implemented in both the lower and middle layer, considering figure 2.1.
- Balancing of the grid can make external control processes necessary, e.g. shutting down the heat pump or leveled curtailment of feed-in based on the peak power of the PV system [54].

In many publications the term of optimal control is used [5, 55, 56, 57, 58, 59]. This mostly refers to perfect data knowledge. The implementation of an associated MPC, however would require some form of prediction. Quality of the control system is directly related to quality of the these predictions, including mostly weather and load forecasts in ES [31].

3 Methodology and Experimental Setup

In the following chapter, the applied data and ES design on which the investigations are based will first be presented. Furthermore some remarks on the component modelling will be made. A third part is concerned with the code basis of this work, including details about some of the utilized Python libraries and the derived code structure. Since only simulations were carried out here, the latter can be regarded as the experimental setup.

3.1 Data Basis

For the following study simulation data from the project *Energetisches Nachbarschaftsquartier (ENaQ)* was used as a basis. ENaQ is linked to the construction of a real neighbourhood on the former airbase in Oldenburg. Here, possibilities for energy distribution between producers and consumers in the immediate vicinity have been investigated, particularly in the sub-project *Physikalische Infrastruktur* [58, 60, 61]. The simulation process that was conducted to determine realistic load profiles, prices and specific emissions is described by Schmeling et al. [61] for the 2017 data, further analysis was carried out by Grimm et al. [58]. Similarly data basis for a second reference, hourly data of 2020, was created within the project. An overview of electricity and heating demand is given in fig. 3.1. Note that the latter represents the sum of seasonal space heating and rather season-independent domestic hot water consumption.

First a reduction in both from 438 to 377 $\mathrm{MW\,h\,a^{-1}}$ resp. 642 to 389 $\mathrm{MW\,h\,a^{-1}}$ for the later year can be observed. This corresponds to a daily consumption of electricity per household of 7.4 kW h...8.6 kW h and heat 7.6 kW h...12.6 kW h, which is realistic for an energy-efficient new-build neighbourhood [62]. It has to be taken into account that 2020 is a leap year and that the number of

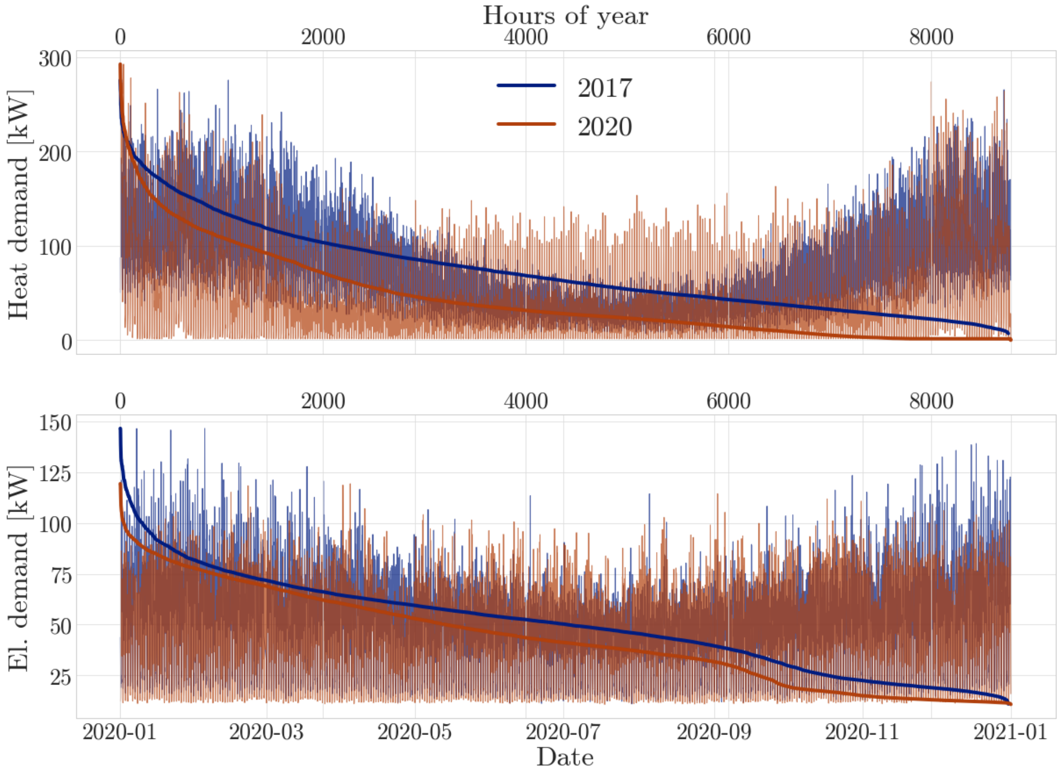


Figure 3.1: Heat and electricity demands and load duration curves for the two years of available data 2017 and 2020.

households is an estimation. An explanation for the significant difference in heating demand can be given with respect to the applied weather data [63] table 3.1. Lower temperature extrema and a mean difference of approximately one degree Celsius, but also the higher overall irradiation were assumed to have an influence on the heating behavior. This is also apparent in the overall structure, as the demand approaches zero even in winter for the 2020 data. In summer, on the other hand, peaks tend to occur here due to hot water supply. However, the electricity profiles are very similar, albeit slightly offset.

Table 3.1: Overview of utilized weather data: ground temperature, direct and diffuse irradiation [63].

Data	**Minimum**		**Maximum**		**Mean**	
	2017	2020	2017	2020	2017	2020
Temperature (°C)	-10.0	-5.8	30.2	33.8	10.1	11.1
DNI ($\mathrm{W\,m^{-2}}$)	0.0	0.0	831.0	794.0	46.5	63.6
DHI ($\mathrm{W\,m^{-2}}$)	0.0	0.0	614.0	536.0	64.4	61.1

In addition to the environmental data above day ahead prices and emission data of the german electricity market are available for the respective years [61]. While the first remains relatively constant (avg. 2017: 34.2 €/MW h, 2020:

30.5 €/MW h) a clear shift of emissions can be seen in figure 3.2 (avg. 2017: 0.53 t/MW h, 2020: 0.41 t/MW h). Both variables also show some degree of correlation. This is plausible, given the fact that a higher share of renewable energies without production costs will lead to lower market prices. Moreover the trend of the German energy mix [1] as well as again the weather difference described above confirm this development.

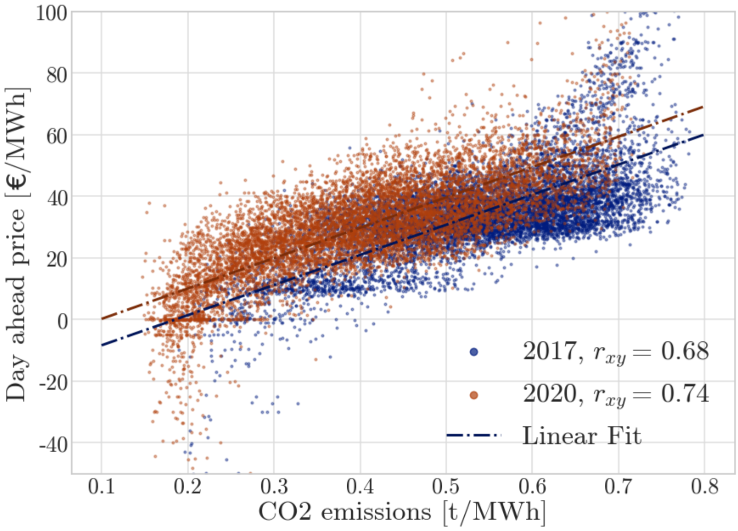
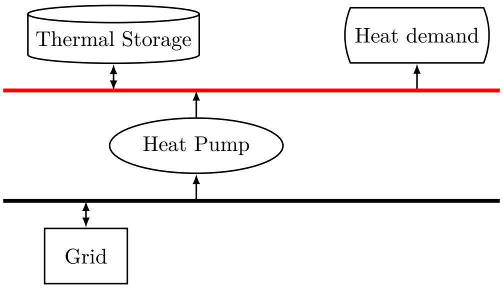


Figure 3.2: Correlation of day ahead prices and specific CO_2 emissions for 2017 and 2020 [61].

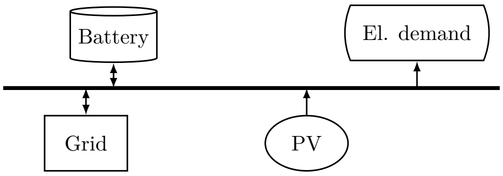
For some of the application cases in chapter 4.3 a further data source was used, containing single household electricity data from 2010 in a higher resolution of one minute provided by Tjaden et al. [64]. Realistic load profiles based on one second measurements can be found here for 74 households, the mean electricity consumption clearly exceeds the above values (avg. $18.1\,\mathrm{kW\,h\,d^{-1}}$). Detailed analysis of the data set can be found in the work of Backhaus [65].

3.2 Energy System Dimensioning and Component Modelling

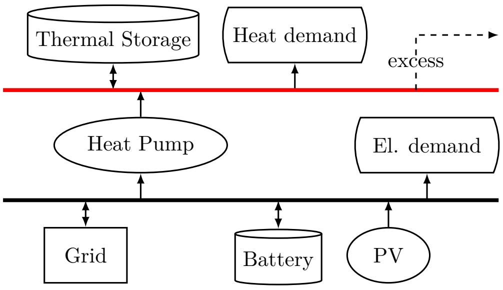
Three energy systems based on the technologies presented in the theory section 2.3 were investigated within the scope of this work. They should represent a selection of realistic applications that can be found in the building and industrial sectors in particular. A heating system consisting of HP, TES, electricity system of battery, PV and grid component, as well as there combination can be found in figure 3.3 in the oemof-solph bus representation.



(a) Heating part system.



(b) Electricity part system.



(c) Full system.

Figure 3.3: Overview of the three energy systems investigated in this work. The full system (c) is assembled of the two sub systems (a) and (b). Based on [61].

The optimisation of dimensioning of individual components as can be found in many other works, e.g. [61, 66, 67] is not dealt with in this thesis. Nevertheless, in addition to the selection of technologies used, this definition will have a major influence on some results, especially the comparison to optimised operation. Therefore choice of sizing is mainly based on an example from the work of Schmeling et al. [61] where a case study on the above described neighbourhood has been conducted. However, it is only one solution for the optimisation of sizing in this paper, further adjustments needed to be made. These are justified below with regard to the modeling of the individual components. PV peak power, the size of the HP and the thermal storage tank were adopted directly ($P_{\mathrm{PV,peak}} = 242\,\mathrm{kW}_p$, $P_{\mathrm{hp,max}} = 400\,\mathrm{kW}$, $V_{\mathrm{TES}} = 20\,\mathrm{m}^3$).

The solar thermal module, on the other hand, was replaced, so that the heat pump acts as the main supplier. Instead, the additionally installed battery ensures greater flexibility of the overall system.

PV and grid component

Very basic models were applied to the PV and grid components in the system shown in figure 3.3. While the first can only act as a source restricted by nominal/peak and solar power, no restrictions are generally assumed for the latter. The grid component will therefore mostly account for the general balance of the electricity system. Similarly heat excess is “allowed” as a further sink in the heat part.

To derive the potentially available solar power input data for diffuse and direct irradiation (sec. 3.1) was simply added up, neglecting the influence of the different orientation of individual modules. A usable power of 200 W m^{-2} was assumed. For the system scale employed, this corresponds to an area of 1210 m^2 which in turn can be used to convert the irradiation into values of available power.

Heat pump

In simplified terms, the HP is merely a converter between electricity part and heating part system. Accordingly different levels of abstraction can be found to describe this transition. As explained in the theory section 2.3, a key characteristic of the heat pump is its COP, eq. 2.13, which is primarily temperature dependent and can also vary with the compressor power to a lesser extent. The easiest assumption would be a constant temperature independent design COP. To first introduce a temperature dependency the Carnot COP method [68] can be used. By defining an efficiency factor η_{hp}, which puts the real design value (e.g. $\epsilon_{\mathrm{h}}(7°\mathrm{C}) = 4.9$) in relation to an ideal Carnot process, a dependency on the ambient temperature can be derived.

$$\epsilon_{\mathrm{h}} = \eta_{\mathrm{hp}} \cdot \epsilon_{\mathrm{max}} = \eta_{\mathrm{hp}} \cdot \frac{T_{\mathrm{max}}}{T_{\mathrm{max}} - T_{\mathrm{min}}} \tag{3.1}$$

Here, ϵ_{max} is the ideal COP from the Carnot process. For the consideration of the actual internal structure of the heat pump, addition resp. substraction of temperature differences might be a useful approximation [68].

A more sophisticated method is the actual component based modelling using the TESPy package (Thermal Engineering Systems in Python) [69]. The full thermodynamical model corresponds to the components of figure 2.7. Now, in addition to the influence of the ambient temperature, part load behavior of the single components can be taken into account for the design range of operating power. To use this model for a mixed integer linear programming (MILP) optimizer as it will be described within the next section, a linearization need to be performed. It was found that fitting with an offset matches the TESPy results very well, especially for lower temperatures. Linearization without offset leads to a significantly less computationally intensive optimization (see later sections), however corresponds less well to the model.
A comparison of those approaches is concluded in figure 3.4 for an exemplary ambient temperature of 7°C. Note that the mx fit resembles the initial COP design value for this temperature. The TESPy characteristic line is shown as well, it will lastly be used for calculation of respective key performance indicators (KPI).

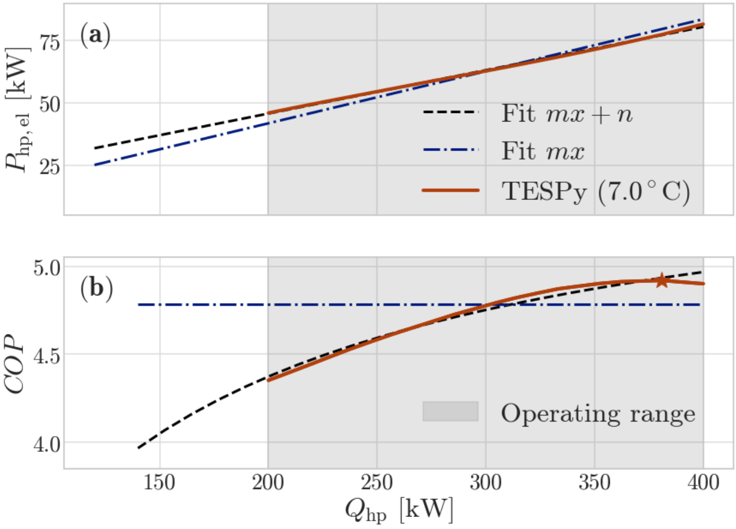


Figure 3.4: Modelling of the HP based on the COP characteristics from TESPy for an exemplary ambient temperature of 7°C. Linearisations correspond to the `OffsetConverter` $(mx + n)$ and `Converter` (mx) components in oemof-solph.

Battery and Thermal Storage

The energy content of the TES can be calculated with respect to equation 2.14, using the above volume, specific heat capacity of water and a temperature difference of 35 K [50], it results in $E_{\mathrm{th}} \approx 815\,\mathrm{kW\,h}$. For the battery, as it is not

included in the example from [61], a total capacity of 500 kW h was assumed. This corresponds to approximately one third to one half of the daily demand. Although there are different guiding rules, the choice ultimately depends very much on the application [70].
One further important aspect is energy loss. A basic approach was adopted here from the generic storage model (`GenericStorage` class) of the oemof-solph package [50, 68], according to which the energy content of a subsequent time step $E(t + \Delta t)$ can be calculated as follows.

$$E(t + \Delta t) = E(t)(1 - \beta)^{\frac{\Delta t}{\tau}} \tag{3.2}$$

However, fixed absolute and relative losses (which could account e.g. for storage ageing) were neglected. A time sensitive loss rate β results in exponential behavior, approximating the relation eq. 2.15. In the case of a single central TES with efficiency class A and a volume of $20\,\mathrm{m}^3$, the maximum permitted heat loss flow is calculated to be 230 W [51]. To take into account different storage levels and additional losses in the overall system, β was estimated at an upper 1% for $\tau = 1\,\mathrm{h}$.

In contrast, self-discharge can be neglected for the battery (based on the Li-ion accumulator, section 2.3) on the charging and discharging time scales under consideration. As battery management is partly included in presented BT control structures, no restrictions are applied here at first. The 500 kW h correspond to a depth of discharge of 100%. Cyclic or calendaric aging were neglected for both components. They might have a considerable influence on the long term operation but mainly on investment considerations.

Later on annual cumulative price C_P and emissions C_E as well as self sufficiency S are calculated as KPIs. As all systems only procure electricity from an external source, those values can be calculated from specific emissions $c_{\mathrm{el,\,E}}$ and specific prices $c_{\mathrm{el,\,P}}$. The amount of utilized energy per time step is denoted by a_el, that of additional grid supply by $a_{\mathrm{el,\,grid}}$.

$$C_\mathrm{E} = \sum_t C_\mathrm{E}(t) = \sum_t c_{\mathrm{el,\,E}}(t) a_\mathrm{el}(t),\ C_\mathrm{P} = \sum_t c_{\mathrm{el,\,P}}(t) a_\mathrm{el}(t) \tag{3.3}$$

$$S = 1 - \sum_t a_{\mathrm{el,\,grid}}(t) / \sum_t a_\mathrm{el}(t) \tag{3.4}$$

3.3 Code Basis

Finally, an overview of the code infrastructure created in the course of this work will be given. There exist a number of libraries and tools in various programming languages for implementing BT and FSM [27, 33]. The Python packages used here should be briefly introduced.

3.3.1 Python Packages

All of the packages listed below were actively developed at the time of writing or shortly before, with appropriate contributions. They are used in other publications and have advantages and disadvantages compared to other possible choices, as will be shown. Detailed information can be found in the respective documentations.

py_trees

In the publication of Colledanchise and Natale [27] a selection of relevant open-source libraries for BT implementations in robotics is presented. It includes *py_trees* as the only tool for usage in Python. The package features built in blackboard (BB), standard sequence and fallback nodes with memory option and provides a run time visualization in unicode format. After initialization of the BB (listing 3.1) read/write access of the respective variables need to be granted in each leaf node [27, 71], as seen in the code 3.2.

Listing 3.1: Initialization of the blackboard using py_trees.

```
def initialize_blackboard(keys):
    """Initialization of read and write access for the
        blackboard"""
    writer = py_trees.blackboard.Client(name='Writer')
    reader = py_trees.blackboard.Client(name='Reader')

    for key in keys:
        writer.register_key(
            key=key,
            access=py_trees.common.Access.WRITE)
        reader.register_key(
            key=key,
            access=py_trees.common.Access.READ)
```

It is noted that py_trees in comparison to other (mainly C++) tools is lower-threshold in terms of programming effort, especially for the application to ROS there exists a variety of tutorials and examples [72]. A major downside of the tool is the lack of asynchronous nodes [27]. In the context of this work challenges arise above all for data input, as the tree needs to be build up recurring with each new data point.
The tree is setup within a function which structures instances of the user defined `py_trees.behaviour.Behaviour` classes, using the introduced node types. An exemplary behavior with blackboard access is shown in listing 3.2. Within an update function, actions can be carried out and conditions defined for the respective feedback. Normally, a behavior also contains initialize, setup and terminate areas for communication with other parts of the system [71].

Listing 3.2: Basic code structure of a behavior for control of the `component` input, using the py_trees package.

```
class Action(py_trees.behaviour.Behaviour):
    def __init__(self, name, component):
        super().__init__(name=name)
        self.component = component
        self.from_bb = self.attach_blackboard_client(name="Reader")
        for key in params:
            self.from_bb.register_key(
                key=key,
                access=py_trees.common.Access.READ)

    def setup(self):
        ...

    def initialize(self):
        ...

    def update(self):
        # perform some action
        self.component.update()
        # feedback
        if cond1:
            return py_trees.common.Status.RUNNING
        elif cond2:
            return py_trees.common.Status.FAILURE
        else:
            return py_trees.common.Status.SUCCESS

    def terminate(self):
        ...
```

Simple comparison of BB variables and a selection of decorator nodes is included in the package [71]. A final tree can be activated, i.e. ticked either singularly or at a specific frequency, py_trees provides the option of so called pre- and post-tick handlers, e.g. for debugging. A sensible tick duration in robotic application lies between 1 ms and 500 ms, this is well achieved in the following experiments.

python-statemachine

For the implementation of FSM the *python-statemachine* package was chosen due to its intuitive handling and visualization options [20]. In principle a class `StateMachine` needs to be created including the definition of states and transitions as well as related actions. An example is shown for the FSM from figure 2.2 in listing 3.3.

Listing 3.3: Basic code structure of the FSM from figure 2.2 using the python-statemachine package.

```
class BasicFSM(StateMachine):
    """Example figure 2.2"""
    s0 = State(initial=True)
    s1 = State()

    a = (s0.to(s1)|s1.to(s0))
    b = (s0.to(s0)|s1.to(s1))

    def before_a(self):
        ...
    def on_enter_s0(self):
        ...
```

An instance of the respective class can then be triggered by an event (here: `a, b`), several transitions can be grouped together within one event. As soon as the corresponding transition does not exist, an error occurs. Actions are performed upon entry or exit of a certain state, e.g. `on_enter_s0`, or before an event [20].

scikit-learn

Scikit-learn is a comprehensive open source library for machine learning in python by Pedregosa et al. [24]. An optimised version of the CART algorithm

(section 2.1.2) can be used in the `DecisionTreeRegressor` class for training of a DT [24].
Implementing DT with if …, else … statements is straightforward, however, for the evaluation they are embedded in the simulation environment using the py_trees package as will be seen later on.

oemof-solph

Optimal control strategies and the concept of model predictive control were already introduced in chapter 2. A tool that is able to handle linear and mixed-integer linear optimization problems is the *oemof* (open energy modelling framework, [73, 74]) package *solph* by Krien et al. [68, 75], which is based on the *pyomo* optimization modeling language [76]. In oemof-solph an energy system can be created using the available set of generic and specific components. Buses for the energy carriers involved are defined for this purpose and connected to the corresponding sinks, sources or transformers [68].
For the optimal design of a system, an iterative process is usually applied, consisting of alternating 1. technology selection and dimensioning as well as 2. optimization of operation. This may include investment costs, among other things.

The assumption of perfect data knowledge over a period of up to one year is of course not realistic, it therefore serves merely as a useful benchmark for the operation of other control systems in the following. A challenge, as described for MPC are time and computational costs in certain system configurations, i.e. the applied models (here especially for the HP). Because of that limitation optimisation data was sometimes split into clusters of several days and processed separately, using the `balanced=True` keyword in the `GenericStorage` class (battery and thermal storage) to ensure correct merging [68].
Further considerations on the aspect of computational speed will be made in section 4.2.2.

3.3.2 Code Structure and Processing

The packages described above form a basis for creating BT, FSM and DT. A prerequisite for their actual application to ES control and further investigations was the development of a suitable simulation framework. Setup and principle structure, including the import dependencies of the decisive scripts is shown in figure 3.5.

Running of an experiment in the `main.py` file requires the access to the simulation function in `simulation.py`. Here also the involved components are defined as instances of the component classes implemented in `components` with there respective abilities (e.g. setting of a power value). The third relevant part is the current control architecture. In case of the BT this involves the initialization of the BB. Different realisations can be created in the `bt/realsisation.py` script using the separately implemented behaviors, which in turn interact with the global BB and control, i.e. edit the respective component instances. By logging their status, output data is generated, that can be further processed (plotting, KPI calculation) with the `post_processing.py` script.

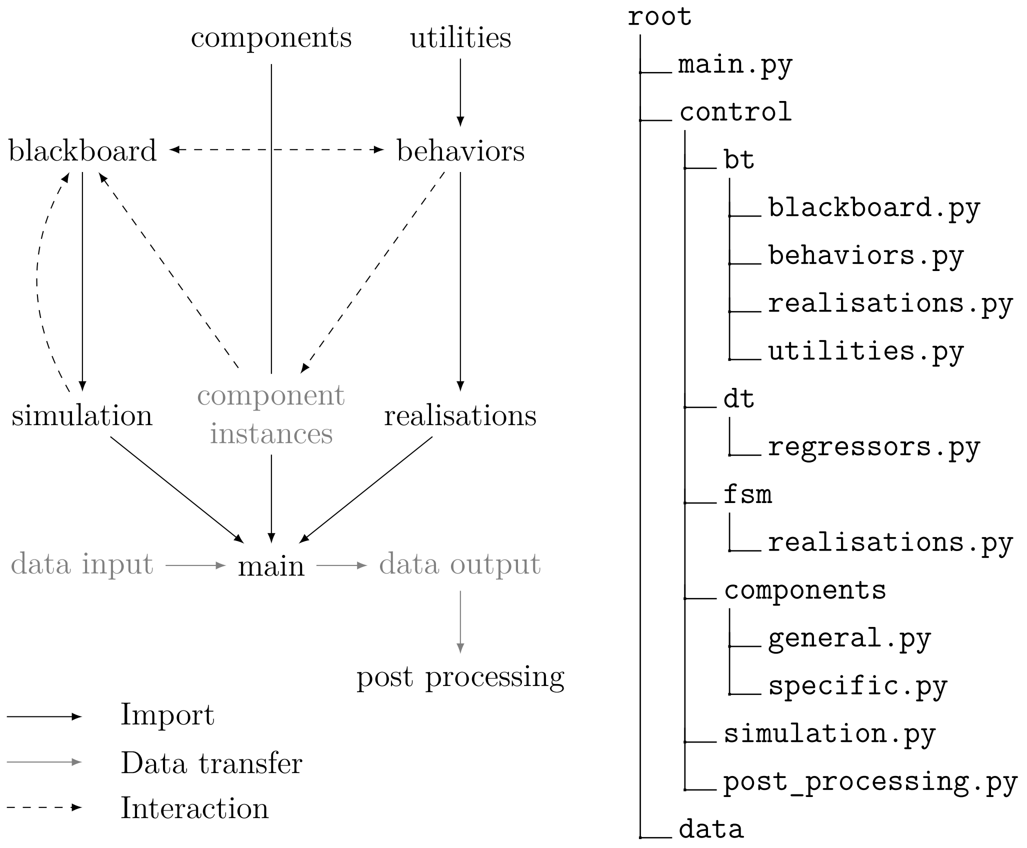


Figure 3.5: Principal code and import structure (a). Relevant excerpt of the code directory (b).

For the FSMs corresponding to the principal structure in 3.3, actions are directly defined within the corresponding class in `fsm/reaslisation.py`. Therefore no additional script exists here. Although a similar storage on a BB would be feasible here, it was not explicitly considered as in the BT implementation. DT are ultimately regressors (see above) that were embedded in a BT-based realization.

Both the BT and FSM realisations need activations at a certain frequency resp. event based in order to perform a control step. One objective for the simulation was not to be limited to the specified data frequency. Therefore, input of a potentially higher step width was processed according to the scheme 3.6. The last existing data point (A) is used to perform a control step (B). Here, the simulation frequency also had to be submitted to the respective storage components, battery and TES, in order to calculate their subsequent energy content.

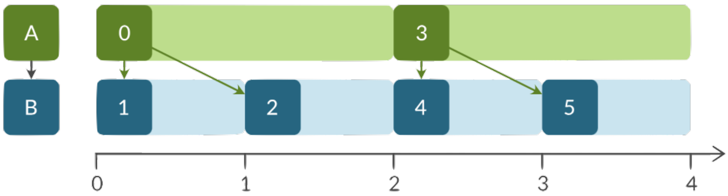


Figure 3.6: Processing scheme. A represents the data input and B the activation, i.e. trigger of the respective control structure. Taken from [77].

As the py_trees library is implemented in a synchronous way, recurring build up of the BT at every simulation time step was required. This is one of the biggest weaknesses of this particular realization. The greatest expenditure of time of approximately 90% was measured here. Depending on the complexity of the tree, the execution frequency for single runs of $\approx 3.5 \times 10^4$ steps (one year of 15 min data) lies between 500 Hz and 1500 Hz[1]. The implementation of FSM, on the other hand, operates two to five times faster.

---

[1] 64-Bit system with Intel® Core™ i7-10850H CPU @ 2.70 GHz and 16.0 GB RAM

4 Results

Overall, the approach of BT in this chapter comprises two major areas of investigation, analysis of the development effort as well as comparison of the operation of different concepts. The first topic is strongly interlinked with the code basis introduced in the previous chapter. Essential control structures for the respective electricity and heat subsystems are derived, using the BT and FSM formulations. For the investigation of the "full system" (see figure 3.3c) component-related expandability is in the focus here.

In terms of operation, FSM should play a subordinate role. As seen in chapter 2, both structures can in principle always be translated into one another, which is confirmed by the shown examples. Instead, optimisation with oemof-solph is utilized as a baseline. Section 4.2 in this context mainly focuses on annual operation and comparison with respect to corresponding KPI.

Application of BT to certain special cases for higher resolution data resp. control activation will be discussed within the last section of this chapter.

4.1 Comparison of the Development Effort

In the theory part, chapter 2, it could be seen that some work has already been conducted to compare FSM with BT. Within the scope of robot operating control the focus is mostly on the expandability of functionalities. When dealing with building or district energy systems, another central aspect is the addition of components. This topic will be examined in more detail in the following section.

4.1.1 States vs. Actions

Before presenting specific control structures, the comparison between them should be justified by some more general considerations. A substantial difference became clear in their definitions in chapter 2. While FSM are based on the introduction of states, BT focus on transitions and feedback of behaviors. This ultimately relates strongly to the way in which such structures are written down, reflecting the manageability of their code based realisation. In principal both can be rewritten into each other to obtain the same operational output, two simple considerations reflect that:

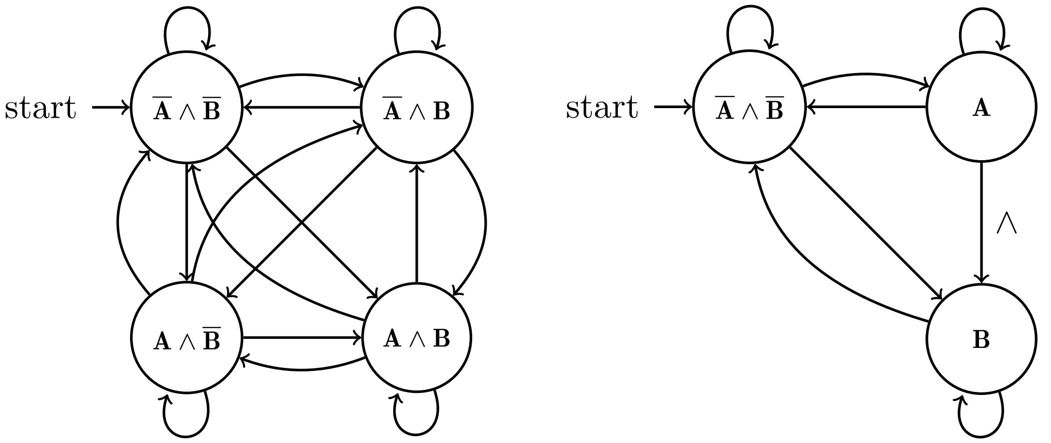
1. A BT is executed from its root. An activation tick may explore the tree as described in chapter 2 and finish one execution step after passing a number of transition, action or condition nodes, again at the root. Equivalently a FSM could be designed in a way, that every state is connected to a "root" state which is reached again after every execution or according to a specified frequency, as e.g. in [36]. Similar structures will be created in the following.

2. FSM on the other hand posses a form of memory, which is reflected in the state that they are in. A behavior tree in turn can be formulated such that a controlled agent is forced to continuously reach a certain leaf node until fulfilment of an equivalent exit condition. For this nodes with memory can be used, sacrificing the advantage of reactivity [9]. Although a signal is in principal travelling through the tree at every execution, the outcome will be the same as the reached behavior acts as a fixed system state.

A first question emerges from these considerations: Based on the structure of a given energy system, how can states or behaviors be derived from which the respective control structures may be composed? One approach to abstracting an energy system was introduced in chapter 3, corresponding to the implementation in oemof-solph. Here, e.g. different energy carriers are depicted as buses in order to connect a number of energy system components in a uni- or bidirectional manner.

Consider the case of a single battery with constant charge power connected to grid. Initially there exist two different states of the system: a charging and an idle state, which might be reached as soon as the battery is full. Several state transitions δ are then feasible, they are usually associated with actions, i.e. "start charging with nominal charging power" or "stop charging". In the bus picture another action would be "draw energy from the grid" (one transition, grid → electricity bus, added to the system) if it is assumed that every component must be addressed at a control level, as will be done in the following. Transferring this to larger ES, the number of states grows very fast with the

number of transitions. More specifically, n connections in the ES graph correspond to $\sum_{k=0}^{n}\binom{n}{k} = 2^n$ states, considering the combination of k connections and a binary distinction between existing versus non-existing flow. For a FSM this means 2^{2n} possible transitions. A discrete definition, based e.g. on subdivision of the power values would complicate this even further [16].
In the example system in figure 3.3c there are nine relevant connections, resulting in 2^9 states, according to this consideration. Assuming that grid supply and usage of the thermal storage are constraint with respect to control of the other components this number can be reduced. However, to achieve an appropriate quantity of states, the following scheme, fig. 4.1b (exemplary for two connections A and B) is proposed.


start
$\overline{\mathbf{A}} \wedge \overline{\mathbf{B}}$
$\overline{\mathbf{A}} \wedge \mathbf{B}$
$\mathbf{A} \wedge \overline{\mathbf{B}}$
$\mathbf{A} \wedge \mathbf{B}$
start
$\overline{\mathbf{A}} \wedge \overline{\mathbf{B}}$
A
$\wedge$
B

(a) 2^n states of with all possible transitions. A power flow is considered either active (e.g. **A**) or inactive ($\overline{\mathbf{A}}$).

(b) Composition of combinations via possible passing through $n+1$ states results in an overall reduction.

Figure 4.1: Distinction between two FSM concepts for representation and control of an overall system state, including the binary flows **A** and **B**.

Instead of separately defining all combinations of active flows, as in 4.1a, they should be composed by subsequent addition. This forces a FSM structure to possibly pass through several states, in order obtain one overall system state at each timestep. Application of this idea will be shown in the following sections.

4.1.2 Heat System

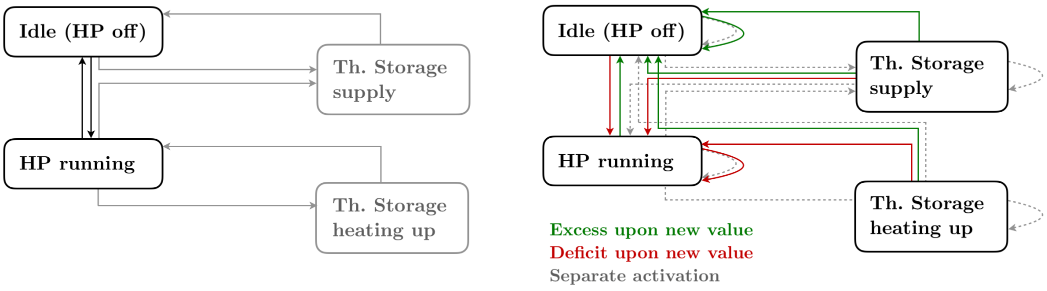
According to the above considerations, a simple heating system, figure 3.3a can be synthesised into five separate flows as shown in the following table 4.1. Here, each flow is connected to a heat value Q_k that should be controllable by the final system.

Table 4.1: Synthesis of the heating system into single flows, definition of control variables Q_k and classification. Q_{exc} is not restricted and can represent a deficit or excess in heat supply.

flow			**boundary cond.**
Heat Pump → Heat Bus	Q_{hp}	≥ 0	constraint
Th. Storage → Heat Bus	$Q_{\text{ts, in}}$	≥ 0	constraint
Heat Bus → Th. Storage	$Q_{\text{ts, out}}$	≤ 0	constraint
Heat Bus → Demand	Q_{dem}	≤ 0	fixed
Heat Bus → Excess	Q_{exc}		unbounded

Given the constraint that demand per time step $Q_{\text{dem}}(t_i)$ is fixed and taking into account a total balance $\sum_k Q_k(t_i) = 0$, three degrees of freedom remain. Note that the picture 3.3a shows two additional transitions, based on the electricity consumption of the heat pump. Since there are no other connections to the supplementary electricity bus, both are only dependent on the control of the heat pump. Therefore they are initially neglected in this section.

A total of $2^3 = 8$ binary states (including no operation as well as "activity" of all transitions) results. It might be suitable to include all of them into a FSM implementation. However, the amount of possible transitions can already make the representation very complicated. Therefore, a different approach was chosen, which is consistent with previous considerations, i.e. figure 4.1b. Based on the idea that the heat or power values of the individual active flows (table 4.1) need to be set for each time step, the FSM must now potentially pass through several states. The following realisations are proposed.



(a) Principle idea of the design. With a fully passive thermal storage the system would be reduced to two states.

(b) Representation of the full actual implementation with additional transitions.

Figure 4.2: Control structure for the heating system, realised with a FSM. Transitions are triggered with respect to the system state in (b).

Essentially, the control system was developed around the two main activities of switching the heat pump on and off, as seen in figure 4.2a. This results in the two main states "off" and "running" which are to be distinguished from the actions performed upon transition. Since with the inherent setting of Q_{hp} the given demand is not automatically met, control of the thermal storage is added in order to supply a potential difference. Turning on the heat pump not necessarily means, that it can full fill all of the heating demand, therefore also "HP running" and "Th. Storage supply" need to be connected. Simultaneous feed-in and feed-out of the storage is neglected here (with regard to modelling by means of a power balance).

In a realistic system, however, control decisions are based on available data, either about the system itself, such as the storage temperature, or on external events, such as changes in demand. Therefore in the actual implementation, represented in figure 4.2b, the FSM is significantly expanded. Here two different forms of activation are distinguished, transitions upon new external data (red and green), as well as separate internal checks with a potentially higher frequency (dashed grey lines). For the first, transitions referred to as "Excess" and "Deficit" initially reflect arbitrary conditions. Examples would be exceedance of fixed storage temperature thresholds, as shown in the next section, but also dependencies on other inputs, as ambient temperature, time of the day or the demand itself. The additional gray transitions become relevant as soon as such conditions are to be checked more frequently than new data is available. This applies in particular to storage restrictions. If they are no longer to be heated or can no longer cover the demand, the heat pump must be switched at short notice.

Another important aspect in realistic systems, that won't be illustrated in the below simulation (because of perfect data availability and functionality), is handling of failure of single components or sensors. The system should at least be able to reach a general error state, e.g. after a defined period of time without new data input or other error messages.

In order to derive a similar functionality structure in a BT, the notation of states was translated into action, i.e. behavior related formulations. Now, triggers that are performed upon entry or exit of states in the FSM (e.g. start charging, set power value) are directly executed within an action leaf node. Only requests and responses are sent via the connections here, which directly handles the task of the fail state described above.

Again, the aim is to consider all components of the system and also to be able to control each heat flow derived in the table 4.1 separately. A resulting general BT structure can be seen in figure 4.3.

The switching behaviour of the heat pump is achieved through two severed sub

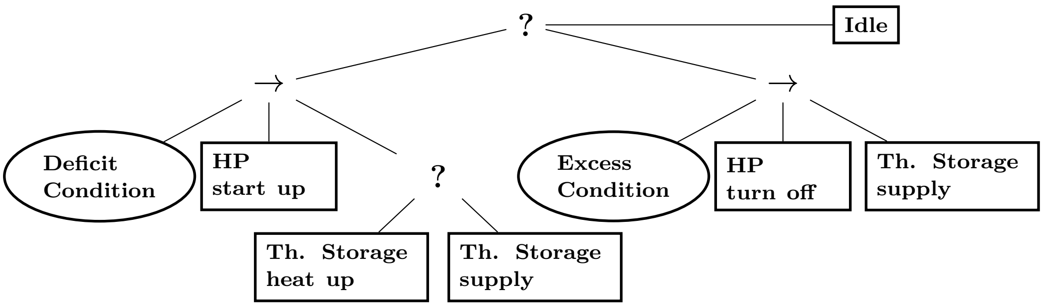


Figure 4.3: Principal representation of a system state based BT control for the heating system.

trees under a selector root node. They are reached with respect to the again initially arbitrary “Deficit Condition”. More precisely, an execution signal remains in the part connected to starting up if this condition is left with success, otherwise the second sub tree will be reached directly. For the control of the thermal storage an additional branch is added in each case. Although present implementations of sub trees can be reused quite easily it is to be noted, that this construction is a bit redundant. If for example, an excess condition is added to the second sub tree (as done in the hysteresis control), an Idle state would also need to be connected to heating up resp. supply of the thermal storage.
In this sense, the realization shown in figure 4.3 could be interpreted as a system state based approach. Another option for segmenting the sub-trees results from the components involved. Connecting the heat pump and heat storage under a sequential node (fig. 4.4) simplifies the structure, but reduces the flexibility of the overall system. Both components are controlled rather independent of each other in this case, which can be advantageous in a simple system where the heat storage is intended to operate passively.

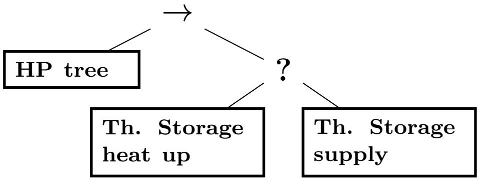


Figure 4.4: Representation of the component based approach for a BT control of the heating system.

Another major advantage of BTs can be seen here: Once a behavior or, connected to that, a component is failing, the execution is automatically re-rooted to the next available option. Cases in which this cannot be foreseen, however, wont be part of the following simulation. Actual operation of both control structures should now be demonstrated for a basic example.

Operation and Illustration of the Modularity-Reactivity tradeoff

A straightforward definition of the deficit condition would be that of a constant threshold temperature within the thermal storage. With this an important difference between BT and FSM implementation without further adjustments can be illustrated. In chapter 2 the modularity-reactivity trade-off has been described. The latter is often identified as one of the main advantages of BT. Figure 4.5 shows such reactive behavior for three exemplary days in 2020.

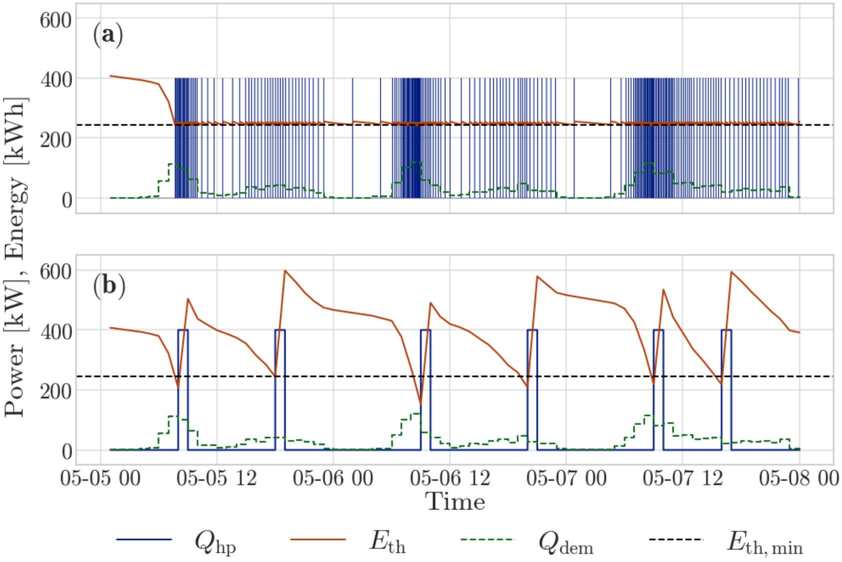


Figure 4.5: Illustration of the HP switching behavior Q_{hp} with a single temperature threshold (corresponding to $E_{\mathrm{th,\,min}}$), controlled with a reactive BT (a) vs. non-reactive FSM (b). The duration between two activations is 100 s.

The implicit objective of keeping the energy content of the storage constant (here at 30%, $E_{\mathrm{th}} \approx 250\,\mathrm{kW\,h}$) has the consequence, that the heat pump in the BT implementation is reacting very fast to its changing environment, although this is not necessarily the desired behavior here. In the FSM on the other hand a state is only left upon the decision making after a new data input, in this case hourly demand Q_{dem}, or due to separate checking of a side condition, e.g. undercutting the lower limit of the storage. Therefore a, in this case, much more reasonable operation is demonstrated in fig. 4.5b. Of course, there are several options to align the output of the two structures, some will be discussed in more detail later on:

- Make the FSM reactive: all state transitions are possibly used at every timestep,

- Execution of the BT only upon new data input ($\Delta t = 1\,\mathrm{h}$),
- Creation of an upper “Excess” condition, i.e. second threshold for the BT, as shown in figure 4.3.

For the first two approaches operation of both systems is shown in figure 4.6, once at an execution time $\Delta t = 1000\,\mathrm{s}$ and a reactive FSM as well as once with the data frequency of $\Delta t = \Delta t_{\mathrm{data}} = 3600\,\mathrm{s}$. The principle structure corresponds to that of fig. 4.5. While detailed structures cannot be made visible for a date range of one year, the decisive fact that the difference is zero, i.e. both control systems work exactly the same in both cases, should become clear.

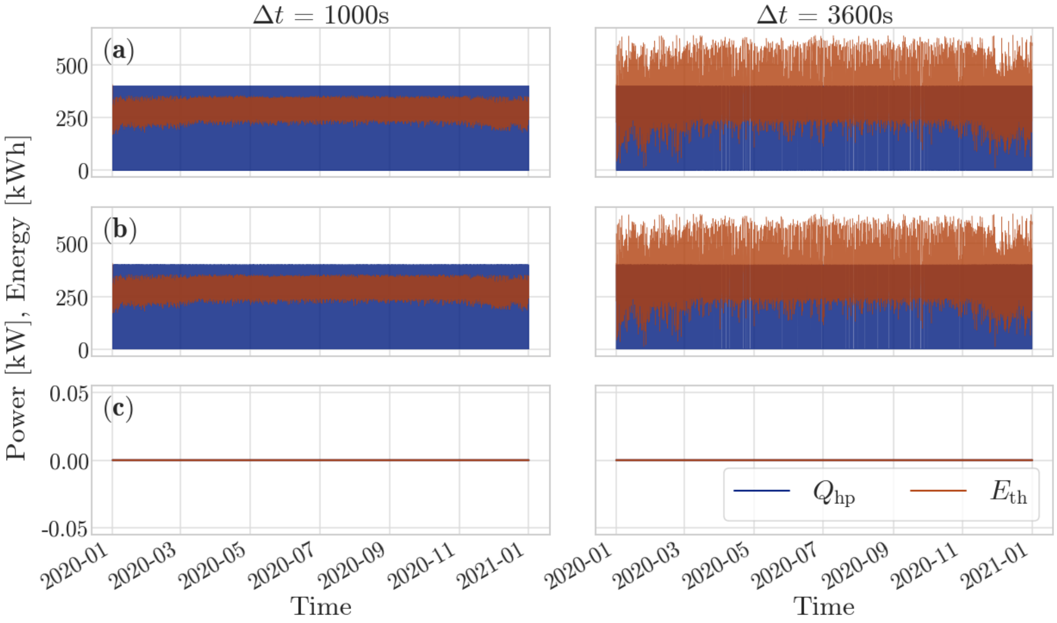


Figure 4.6: Exemplary operation of the FSM (column (a)) and BT (column (b)) with different execution frequencies. For a reactive version of the FSM both systems work equivalent, as seen in the signal difference (c).

One aspect, that has been mentioned but not tackled for the implementation of the FSM (since it was also not further used for KPI analysis in section 4.2) is the attention to a grid component. In correspondence to the above considerations, it could generally as well be handled as a buffer, that is passively updated. For illustration as well as simulation purposes the control of the grid should nevertheless be integrated into all of the presented BTs.

This is simply done by adding a second electricity power balance to the system, that is increased only by the demand of the heat pump and equalised in an additional grid leaf node or sub tree as seen in figure 4.7.

An important tool for the realisation of the mentioned power balances is the utilization of a BB, which is often named as such in connection with BT. However, it does not represent a fundamental advantage, as the implementation of a key value memory is of course also possible for FSM. The overall picture of a control structure for the heating system with a BB would therefore look as follows.

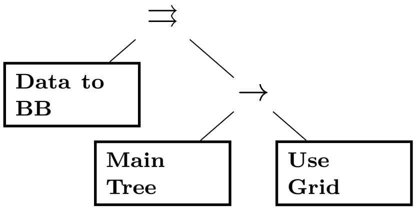


Figure 4.7: Adding of a data behavior and grid component to the main BT realisations shown in figure 4.3 and 4.4.

4.1.3 Electricity System

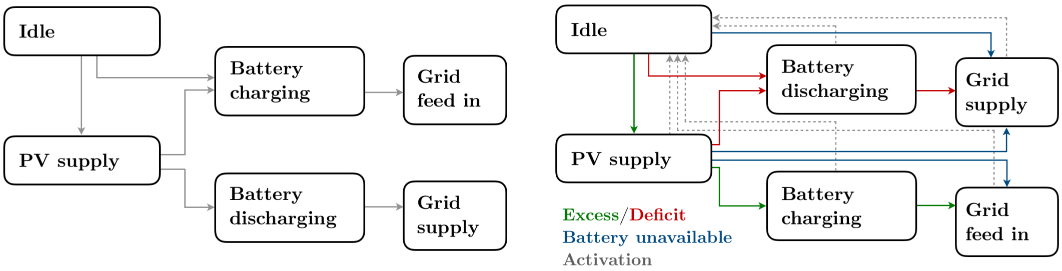
The approach of developing a control structure for the ES 3.3b is essentially the same as in the previous section. Instead of only allowing a passive excess as in the previous example, the aim and the main difference is now to also actively address resp. control the bidirectional grid component. It will be seen, that this not only results in a larger number of states but also creates further restrictions in the use of the FSM.

An overview of the energy flows synthesised from the representation in figure 3.3b is shown in the below table 4.2 and the basic control idea in fig. 4.8a. Note again, that in principal combinations of all flows could be grouped as states of the system, to create a very different formulation.

Table 4.2: Synthesis of the electricity part system into power flows P_k and classification, with respect to figure 3.3b.

flow			**boundary cond.**
PV → El. Bus	P_{pv}	≥ 0	constraint
Battery → El. Bus	$P_{\mathrm{bat,\,in}}$	≥ 0	constraint
Grid → El. Bus	$P_{\mathrm{g,\,in}}$	≥ 0	unbounded
El. Bus → Battery	$P_{\mathrm{bat,\,out}}$	≤ 0	constraint
El. Bus → Demand	P_{dem}	≤ 0	fixed
El. Bus → Grid	$P_{\mathrm{g,\,out}}$	≤ 0	unbounded

Again, meeting the fixed demand P_{dem} is set as the overriding goal. Every flow from the table is reflected in a state. Furthermore an “Idle” state need to exist for the event that there is neither demand nor production. Although the figure suggests a prioritisation which also appears in section 4.2, a very generic structure should be created initially. If available, PV will be mostly used, but also the direct battery or grid supply should be possible. Therefore the conditions for the so called “Excess” and “Deficit” transitions in figure 4.8b might be chosen very differently to obtain different behavior of the system.



(a) Principle idea of the design. Prioritized use of PV over battery and grid.

(b) Representation of the full actual implementation with additional transitions.

Figure 4.8: Control structure for the electricity system, realised with a FSM. Transitions are triggered with respect to the system state in (b).

Running different components at the same time was identified as a major challenge in the here presented control with FSM. Consider the operation that was called non-reactive in the previous section. For actions that should not be executed with high fluctuation, as e.g. battery charging and discharging, this operation is definitely sensible and a reason why combinations of FSM and BT has been proposed in [30]. In the example, however, additional actions should be carried out as soon as the charging rate is exceeded or the remaining capacity of the battery is no longer sufficient. In addition to activating the battery, one of the grid states of the FSM would have to be reached. In those states, inspections of the battery status are formally not possible, although they may be necessary to end the charging process at some point.

This problem is reflected in the simulation results shown in figure 4.9. Here 13 h of operation for a night period, i.e. predominant grid and battery supply are shown. After the first time step (hourly resolution) no PV production is possible, therefore the battery starts discharging. Since it cannot full fill the entire demand on its own, due to a low charging rate, additional grid supply is needed. The execution period of both BT and FSM ($\Delta t = 1000\,\mathrm{s}$) is smaller then the data rate. Therefore in the non-reactive FSM only one discharging step is simulated before the system switches into the grid supply state. It only leaves this state upon input of a new value which leads to an unrepresented discharging. As soon as the storage level reaches 20%, use of the battery is skipped in the BT. If this level would be reached by the FSM, operation would again be synchronous.

As said, in a realistic system, the action of entering the battery state is associated with a process or trigger to start charging. The first does not necessarily need to be aborted when leaving the state, however, the difficulty in continuing to track the component status (if not in the respective FSM state) is still apparent here.

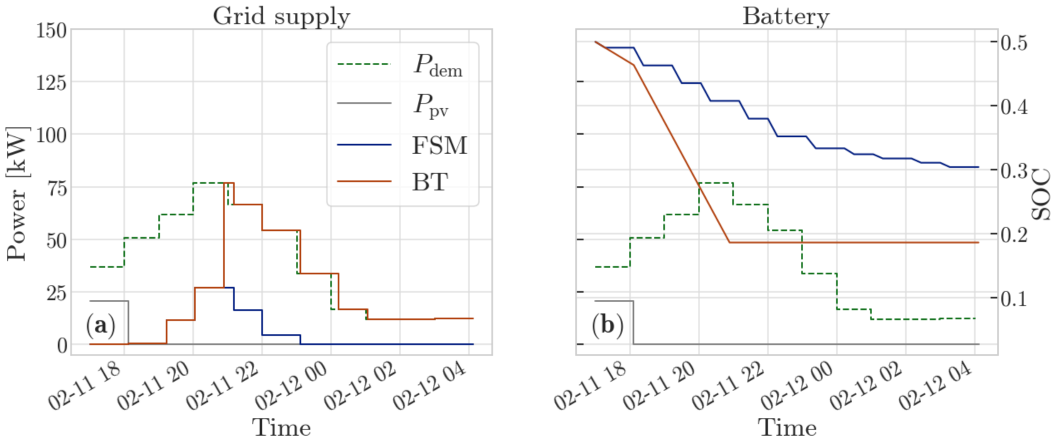


Figure 4.9: Grid supply (a) and battery *SOC* (b). Comparison of non-reactive FSM and BT operation with activations $\Delta t = 1000\,\text{s}$. Very low C-rate of 0.1 for illustration.

As a solution, when setting up a control structure in the here proposed way, it is inevitable to run it reactively, i.e. changing the system state at a higher frequency instead of only upon new values. In the simplest case, the FSM returns into the idle state at every activation in order to make new decisions, see figure 4.8b. This, however corresponds exactly to the operating mode of a BT, for which the basic idea is shown in figure 4.10.

As in the previous section an implicit overall arrangement of sub trees under a fallback node was chosen for the principle construction. Those selector sub trees also represent general excess resp. deficit states of the overall system. As discharge and overcharge protection play a much more significant role for electrochemical storage, i.e. Li-ion accumulators, safety conditions are itemised individually in the shown figure.

Based on the sensible prioritisation of PV energy production, this component takes a subordinate role here. Instead of defining a condition first, now a PV behavior or sub tree is reached at the beginning of each excecution of the tree. In correspondence with the FSM representation an idle state was added to

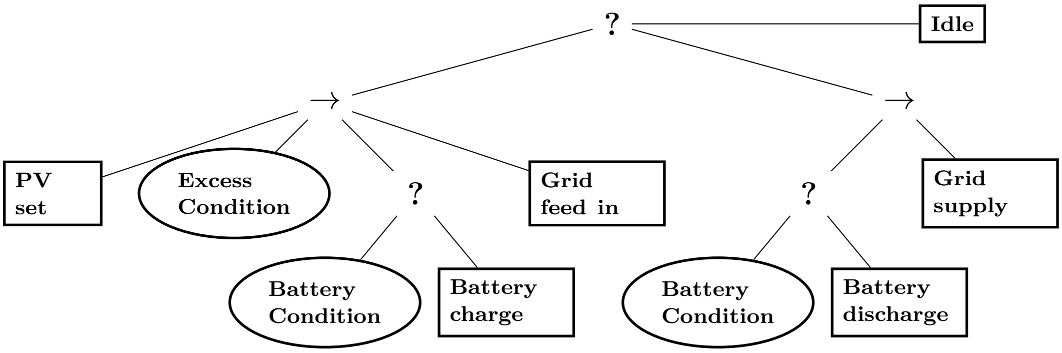


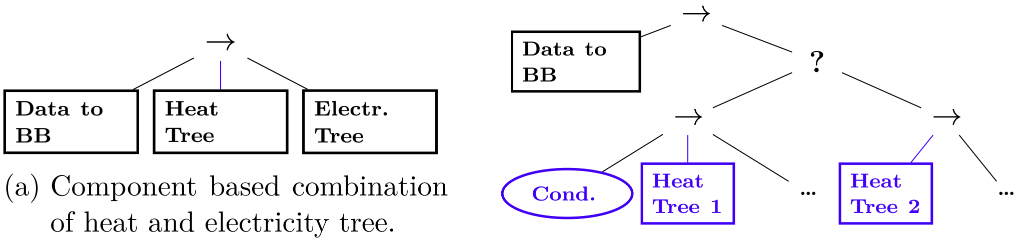
Figure 4.10: System state based realisation of the electricity system control. Idle node is only reached if no demand and no PV power exist.

the root of the tree, which is reached only after failure (e.g. no demand and no irradiation) of all other branches. Through this design, the idle state does not have to be reached after every execution as in the reactive implementation of the FSM. Note, that again a blackboard is used for interaction of single behaviors with the overall power balance, compare figure 4.7.

The control structures finally obtained in this section, i.e. reactive FSM and BT where again able to operate in the exact same way. For test runs, an excess condition $P_{\text{pv}} > P_{\text{dem}}$, battery condition $0.2 \leq SOC \leq 0.8$ and execution periods Δt of 60 s, 300 s and 900 s where applied.

4.1.4 Combination of Both Systems

The above sections showed, that for two representative simple ES, task switching control structures can be created both using the concept of BTs as well as FSM. They are in principal able to operate in the same way. Comparison will be touched upon in the next section. Here the consequential question of how such systems can be combined will be briefly examined.
In principal two strategies are feasible for the extension. For the heat system electricity demand is generated only through operation of the HP. Therefore one obvious approach would be simple addition of this demand to the electricity system. Consequently, the control structure of the heating system is not affected in this case and the decision-making process is carried out sequentially. Only the electricity system has to handle an increased overall demand within its power balance. A realisation of this strategy using BT is shown in figure 4.11a and the principal idea for FSM in fig. 4.12a. It will be referred to as component based combination in the following.

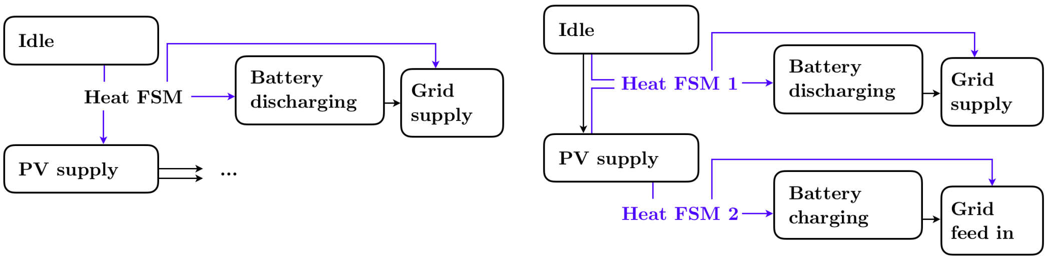


(a) Component based combination of heat and electricity tree.

(b) System-state based combination of the adapted heat and electricity trees.

Figure 4.11: Proposed strategies for combination of the simple system control BTs for control of the full system. Parts of the tree that need to be edited are marked in purple.

For improved communication between components and overall operation, this is certainly oversimplified. Therefore as a second approach system state based combination of the single control structures seems to be favorable. A surplus of available PV power could, for example, favor the use of the HP, while this would not have been switched on with the simple heating control under the same system conditions (storage level etc.). To achieved this, editing of the heat tree and insertion into the respective subtrees of the electricity system control is feasible, as seen in figure 4.11b. For the FSM, however a significant effort of restructuring is expected for this scenario, whose principal design is outlined in figure 4.12b.



(a) Insertion of the heat FSM into the electricity FSM.

(b) Insertion of adapted heat FSM into the electricity FSM.

Figure 4.12: Component based (a) vs. system state based approach (b) for combining the FSMs. Purple lines illustrate added transitions.

Note that the figures for the FSM merely represent schematic structures. The insertion of the heat system control consisting of four states itself results into one transition from the idle state into the initial state of the heat FSM, but requires up to four transitions each for exiting into the “PV supply” resp. “Battery discharge” and “Grid supply” states.

Conclusion and Metrics

Cyclomatic complexity and GED were introduced as metrics for the comparison of graph based structures in section 2.2. Calculation of the first is suitable for an evaluation of the increase in complexity of FSM. It is elevated from $CC = 15$ in both the simple systems to $CC = 23$ in their component based combination.

For calculation of the GED the Python package NetworkX was used [78]. The results for the expansion of the BT are clearly traceable. While in component base combination only one transition has to be added ($GED = 1$), at least two nodes will be edited and two transitions added in the system-state based combination, resulting in a GED of 4. For the FSM, figure 4.12a, the GED is calculated as 13. The system-state based combination of FSM has not yet been realised.

To a certain extent, these figures can be linked to the coding effort required to expand the respective structures [36]. As mentioned in section 2.2, however, the choice of tools always takes on an important role. At this point it can therefore only be stated that the extension of the BT structure is easily possible using the py_trees package. For the component based combination not a single line of code hat to be deleted. The expansion of the FSM so far was associated with a significant amount of additional work in code adaption.

4.2 Investigation of Operation

For all systems, basic control structures were created in the previous section. Conditions and behaviors are formulated there in very general terms or not at all. In addition to the prioritization generated by the structures themselves, operation can now be influenced in particular by refining, introducing and also deleting sub trees. In the subsequent comparison, DTs and classical control patterns are introduced and combined in an attempt to exploit the BT structure in more detail.
The systems under consideration are very simple. At best, the composition to more complex control structures using the knowledge for simpler systems might be advantageous.

4.2.1 Optimisation

To compare annual operation of different control systems, appropriate indicators must first be specified. The following considerations focus on the comparison of self sufficiency, CO_2 emission and (static) price. Optimisations in oemof-solph are based on the definition of costs for the different energy flows, as explained in section 3.3.1. Accordingly, to obtain optimality with respect to the above objectives, those costs had to be adjusted in each case. Investment costs were not initially taken into account, it was therefore assumed that PV and battery inflow and outflow costs were always equal to zero. Grid feed in and supply costs has been adapted according to the following table 4.3.

Table 4.3: Costs applied to the optimisation problem for the different objectives. Based on section 3.1, prices according to the EEG [2].

KPI	year	grid supply	grid feed in
1. Self sufficiency	2017	const. > 0	0
	2020	const. > 0	0
2. CO_2 emissions	2017	0.190...0.781	0
[$\mathrm{kg\,kW\,h^{-1}}$]	2020	0.147...0.729	0
3. Electricity price	2017	29.28	-10.69
[$\mathrm{ct\,kW\,h^{-1}}$]	2020	31.81	-7.54

Maximization of self sufficiency is equivalent to minimization of total electricity consumption, see equation 3.4, and therefore total monetary costs in the case of the basic heating system. Because there is no possibility of own production, self sufficiency will be zero in this case.

Results of the optimisation for all three systems (2020) are concluded in figure 4.13. The degree of self-sufficiency is not indicated here as it is constant for the electricity system (2020: 83.6%) and fluctuates only slightly for the overall system (2020: 77.0%...77.9%). It can be seen that the load duration curves can differ considerably from one another, whereby self sufficiency and emission optimisations are more likely to be similar in this form of presentation.
The electricity consumption of the heat pump can be calculated either using the linear approximation, that was applied in the optimisation or based on the exact nonlinear COP curves obtained from the TESPy model, see figure 3.4. Both are shown in the figure, however, the difference is very small (heat system: $\lesssim 1\%$, full system: $\lesssim 2\%$) for all cases considered. The TESPy model was also used for calculation in the simulations of the other control concepts.

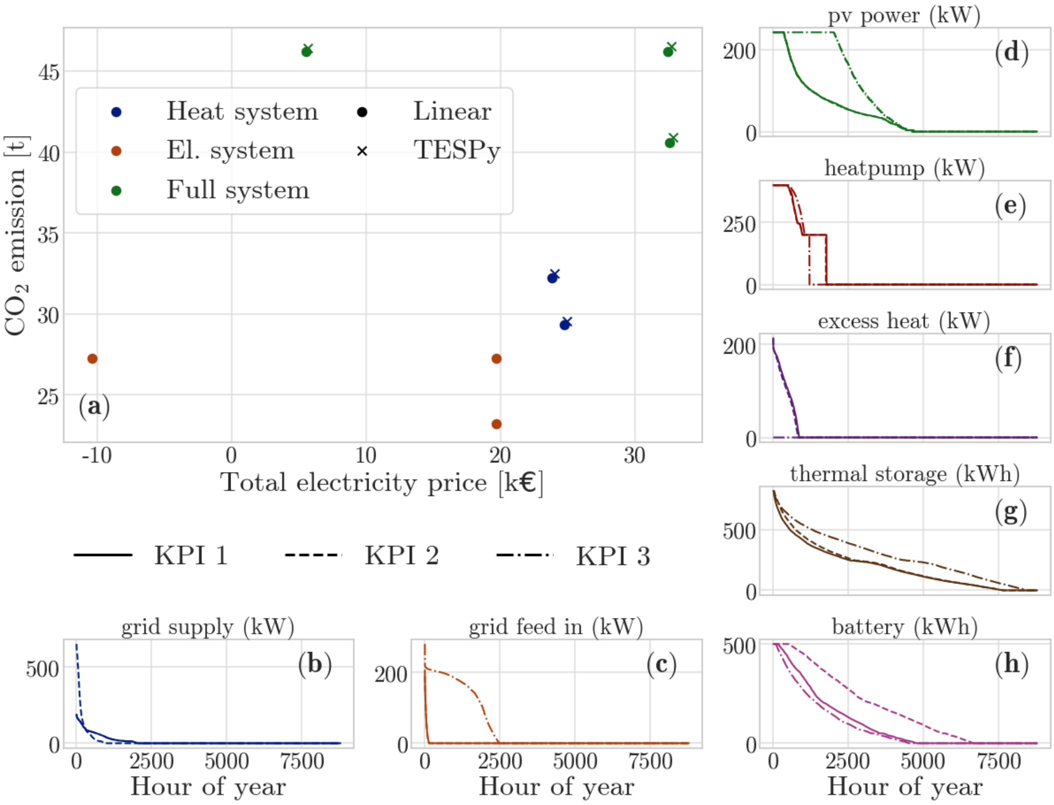


Figure 4.13: All optimisation results for 2020 (a), with comparison between TESPy vs. linear HP modelling. Comparison of important load duration curves (b) - (h) for the three different optimisations with respect to KPI from table 4.3.

Only for the electricity system single optimisations could be performed over the course of the respective full year. For the inclusion of the heat pump, data slices needed to be processed separately for reasons of computational time constraint. Details are provided in the next section.

4.2.2 Considerations on the Computation Time

One fundamental difference between the optimisation and the BT concept is their initial interpretability. A similar situation showed to apply for their computation times, which fluctuated extremely for the optimisation. As explained in section 3.3, BT execution, i.e. simulation speed is restricted by the implementation, hence expected to be relatively well predictable. For the application in real systems computational effort is seen as an important factor, which should be included into the decision for one or the other control strategy.

Optimisation of the electricity system control in oemof-solph takes $\lesssim 1\,\mathrm{s}$ for annual hourly data, which is considerably less then run time of a single BT simulation of approx. $\lesssim 10\,\mathrm{s}$. For the other systems, however, modelling of the heat pump had a major impact on the execution time. To show this, exemplary runtime measurements were conducted for the full system and 2017 hourly data. The results comparing the use of an offset model, offset model with additional start up costs and simple converter model (see section 3.2) can be found in figure 4.14a.

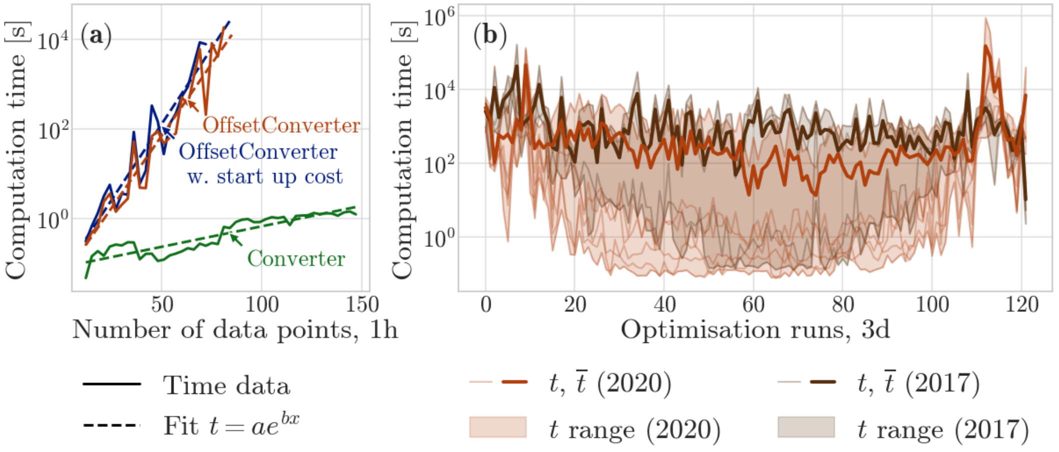


Figure 4.14: Comparison of computational effort of the optimisation in oemof-solph. Application of different HP models (a) and averaged computation time using only the `OffsetConverter` for batches of 72 data points (3d) in the course of one year (b).

It can be seen, that computation time is increased exponentially for all three models. As the `OffsetConverter` should be used for the following investigations, subdivision of the annual optimisation proved to be necessary, considering run times of up to $1 \times 10^4\,\mathrm{s}$ for less then 100 data points. An overview of averaged optimisation runtime $\bar{t}$ (three day intervals $\hat{=}$ 72 data points) for the construction of annual result is provided in figure 4.14b.

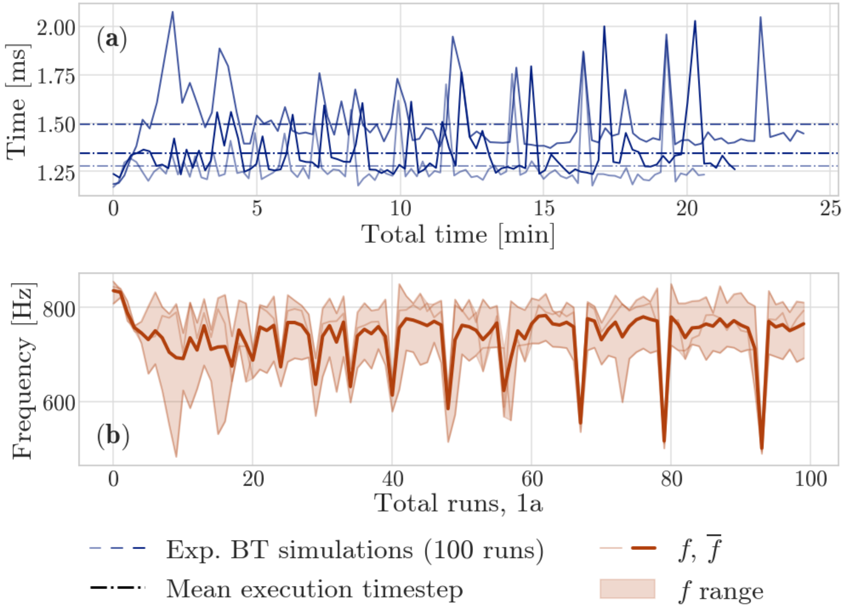


Figure 4.15: Development of BT execution time (between two ticks) (a) and execution frequency f (b) for different sets of subsequent annual simulation runs.

Instabilities also occurred in a larger number of consecutive BT simulation runs, but on a much smaller scale. In figure 4.15 computational performance of a BT control for the heating system is shown. Each run corresponds to simulation of one year of hourly operation. For the first 100 subsequent runs, average computation time of one activation of the tree does not considerably exceed $\Delta t = 2\,\mathrm{ms}$[1]. Concluding, average simulation times (hourly annual operation) between 11 s and 13 s for the BT investigations stand in contrast to average optimisation times of 97 h (2017) and 62 h (2020) for runs including the offset HP model.

---

[1] For comparison, according to the py_trees documentation [71], tick rates of $\Delta t = 1\,\mathrm{ms}$ to $\Delta t = 500\,\mathrm{ms}$ are recommended for real time ROS applications.

4.2.3 Comparison to DT, Classical Control and Optimisation

Before discussing comparative results in the form of annual KPI, some aspects concerning the implementation of DT and hysteresis control structures should be briefly reviewed on the basis of the heating system.

Hysteresis as BT

For the formulation of a simple hysteresis control BTs are certainly overcomplicated. However, since the simulation framework is designed for BT and extensions are planned, the following implementation 4.16 is appropriate. When using the component-based approach, it looks as follows:

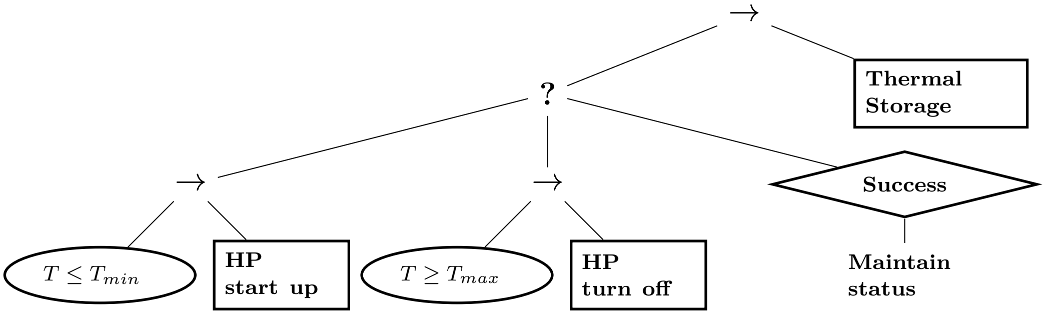


Figure 4.16: Component based approach to hysteresis switching of the heat pump.

As soon as one of the lowest level conditions returns a success feedback, the respective HP action is performed. For a storage level withing the boundaries current operation is to be maintained. The “Maintain status” node is reached in this case and delivers a success (using a success decorator) before the passive TES control is realized within the second sub tree “Thermal Storage”.

DT as BT

For DTs there exist very mature training algorithms, as seen in chapter 2.1.2. In an actual control system on the other hand the direct application and addressing of the single components is rather difficult. After each activation, only one leaf node is reached in which all relevant actions must be carried out. Instead of realising a separate environment for operation of DT, the framework of BT as introduced in chapter 3.3 can be used, equipping only the relevant behaviors with a decision sub tree. This idea is illustrated in figure 4.17. Here start up of the heat pump is solely controlled by the decision of the respective DT regressor. Supply of the adjacent components is taken over by the BT in consideration of the power balance.

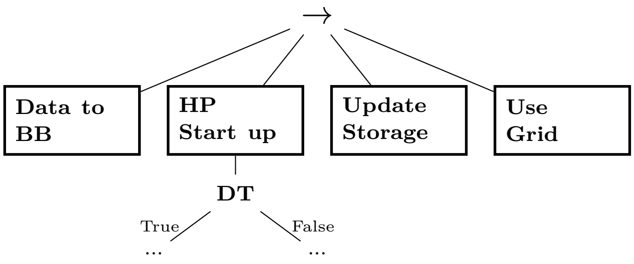


Figure 4.17: Decision tree implementation using the BT framework.

In the shown tree no further boundaries are respected and provision of the demand cannot be guaranteed as seen later on. One option to avoid this is the introduction of safety thresholds, such that the decision making is limited with respect to demand and actual storage content, i.e. temperature, eq. 4.1.

$$Q_{\mathrm{hp}} = \min\left\{Q_{\mathrm{dem}} + \frac{E_{\mathrm{S,\,tot}} - E_{\mathrm{S}}}{\Delta t},\ \max\left\{Q_{\mathrm{hp,\,DT}},\ Q_{\mathrm{dem}} - \frac{E_{\mathrm{S}}}{\Delta t}\right\}\right\} \tag{4.1}$$

Another option, particularly illustrating the advantage of BTs is the direct insertion of a DT into a control structure that ensures coverage of demand, e.g. the above hysteresis control. In this case the heat pump is definitely started or turned off upon the respective condition, while the exact power output is defined by the DT.

Of course, those considerations are not limited to the heating system. Also in the structure fig. 4.10, for the electricity system (and due to the expandability of BT also for the full system), DTs can be applied. In the first case this might be particularly worthwhile for charging of the battery, since available PV power will mostly be used and the grid component rather takes on the function of a buffer. A sensible selection of training data is required. Here, regression features can be storage content, demand, time of the day, solar power and ambient temperature but also day ahead price and specific emissions if available for the system. An exemplary training of a DT on 2017 optimisation results for the heat system is shown in figure 4.18.

It can be seen, that DTs are able to fit the optimisation relatively well. The correlation of CO_2 emissions and day ahead prices, seen in chapter 3.1, favours

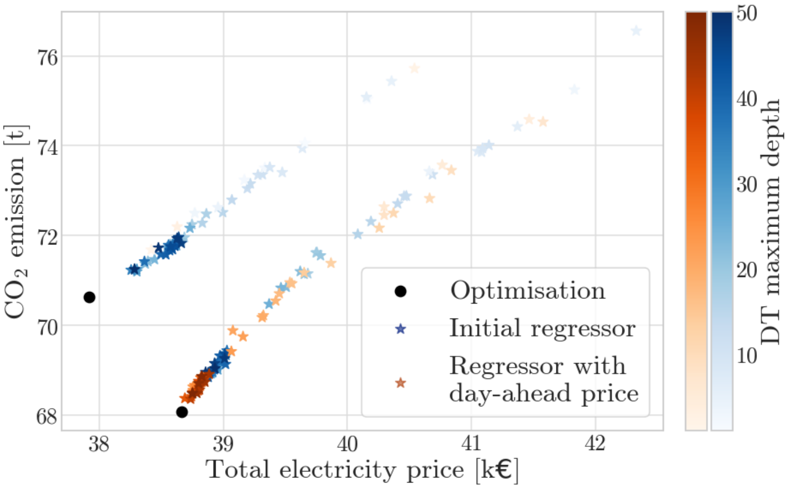


Figure 4.18: DT training on 2017 heat system optimisations. For the initial regressor, heat demand, storage content, ambient temperature and time of the day were applied as features. An additonal introduction of day-ahead prices is favorable for approaching the emission optimum.

the introduction of the latter as feature for approaching the emission optimum. Although, it is to be mentioned, that for the shown data points mostly the demand is not completely meet. Deficits in the range of 0.02% correspond to approximately 100 kW h in the shown case. A sensible value for the maximum depth was determined to lay between 10 and 30 decision layers.

4.2.4 Results

According to the research question of this thesis, first, all of the above approaches should be compared with the optimisation for the basic systems. Training of the DT was performed with respect to the optimisation results of 2017, note the considerations of the previous section and figure 4.18. All further results relate to operation for 2020.

Heat System

In a first step hysteresis control was simulated using the described BT realisation, fig. 4.16. The lower temperature bound was varied between $T_{\text{min}} = +2\text{K}$ and $+20\text{K}$ and upper bound between $T_{\text{max}} = +4\text{K}$ and $+22\text{K}$ compared to the storage reference temperature (with $T_{\text{max}} \geq T_{\text{min}}$). Due to the hourly resolution of activation not all of those limits are suitable to fullfill the heat

demand completely. A smaller range of +4K...+7K lower and +7K...+15K upper temperature bound was therefore chosen for further extensions.
When directly applying the DT approach from figure 4.17 without safety margins, heat deficits for regressors with a depth between 10 and 30 appeared to be unavoidable. Two solutions were investigated to avoid these deficits, a combination of DT regressor and hysteresis control, as well as bounded DT decision according to equation 4.1. The ten simulation runs associated with the lowest CO_2 emissions for each case are shown in figure 4.19. These also include the runs with the lowest annuity, showing that there is no real sensitivity to one or the other parameter. Rather, the relation is approximately linear for the majority of data points.

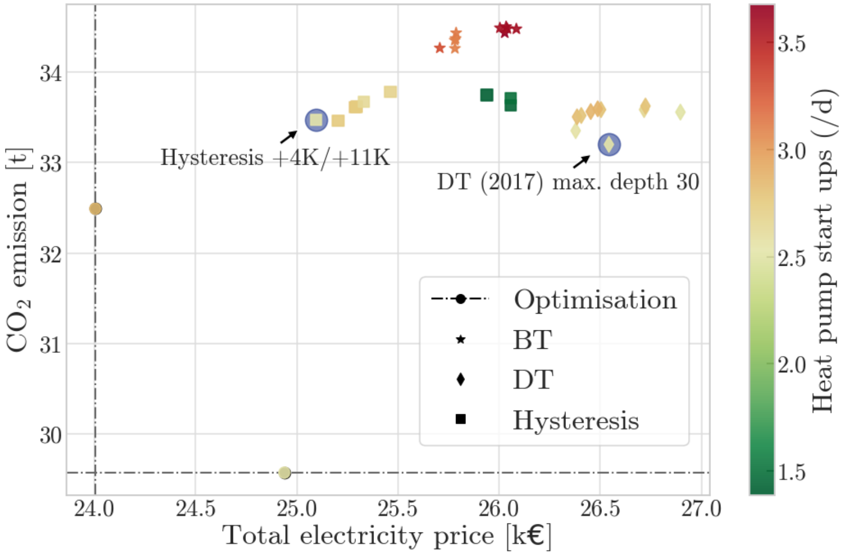


Figure 4.19: Comparison of the ten runs with lowest CO_2 emissions for each case and optimisation baseline. Here the combination of hysteresis control and DT regressor is refered to as BT. Note that start up costs are not included into the calculation of the total electricity price and not considered in the optimisation.

Introducing a DT regressor into the hysteresis control shows no advantage over simple switching between maximum and zero power within certain TES temperature thresholds in the HP. It is to be noted, that the results are generally very close to the optimal operation[2]. Only the restricted DT with a maximum depth of 30 yields slightly lower emission when compared to the hysteresis control.

---

[2] As described in section 4.2.2, optimisation results are obtained through slicing and subsequent reassembly of the data. If it had been possible computationally to optimise the entire period in one step, the results would probably be further apart. This is because the optimisation is “forced” to a certain (possibly non-optimal) storage status at certain times, in this case every three days. For all other control structures, this does not apply.

A further aspect that could be decisive for the choice of a control system, is illustrated in the color range of the figure. It can be seen that the average daily number of start ups of the HP is significantly reduced for certain variations of the temperature thresholds, while the total electricity price is only increased by approximately 5%. Assuming installation costs of 250 €/kW and a medium lifecylce of 25 a for three start ups per day [79] (neglecting other ageing factors), the influence can be roughly estimated, as seen in table 4.4. The potential savings through increased lifetime of the component (1.91 k€) corresponds approximately to the additional costs from electricity consumption (1.94 k€) in the example and year considered.

Control	**Start ups [/d]**	**Add. costs [k€/a]**
Optimisation (Price)	2.82	3.76
Hysteresis +4K/+21K	1.39	1.85

Table 4.4: Estimation of additional costs for two operation strategies of the HP. Here wear of the compressor through start up of the HP is assumed to be the only influencing factor on the total life span of the component.

For a sensible comparison, however, optimisations should also be carried out taking into account the start-up costs directly.

Electricity and Full System

For the electricity system a similar control strategy should be implemented, as the basic prerequisites are very much alike. The battery has a lower and an upper limit and the grid acts equivalent to the heat excess as a buffer. However, grid feed-in costs were defined to be zero with respect to emissions, they are remunerated, influencing the annuity. It is therefore generally not meaningful to restrict the PV production.

Instead a value that can be controlled by a separate DT is the charging power of the battery. This was done, again using a regression of the 2017 optimized data with the available PV power as an additional feature. Furthermore, a priority based approach for the BT, figure 4.10 was simulated with battery conditions $0.0 \leq SOC \leq 1.0$, resp. $0.2 \leq SOC \leq 0.8$. The results for the electricity system operation are concluded in figure 4.20.

Surprisingly, the classical priority based control has an equal annual price KPI compared to the optimisation. At the tighter limits of the battery SOC, it is even below the values of the optimization. This can of course be explained

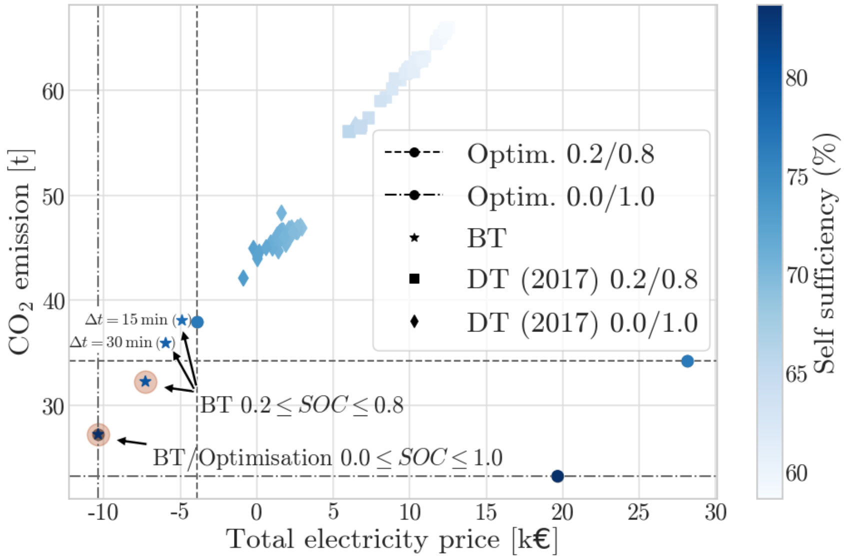


Figure 4.20: Priority based control (BT) and introduction of a DT regressor for charging in the electricity system. Comparison to the optimisation with different battery thresholds. The runs with modified activation frequency (other than 1 h) are placed in brackets.

by the hourly resolution. While the optimization is able to set precise limits, an action in the tree is only adjusted if these limits are exceeded (the conditions are ultimately violated in this setup). By increasing the BT activation frequency, the price optimum is approached again. This can be seen for $\Delta t =$ 30 min and 15 min in figure 4.20. The DT regressor, on the other hand, is far above the values of the optimization in the examined configurations and in all aspects.

Finally control structures can be combined according to the considerations in section 4.1. This was, however, done here only for the component based approach figure 4.11a and the best operating structures from the subsystems. Consequently, a decision is always made in favour of the heat pump first. The electricity BT then solely operates with an adjusted power requirement, i.e. increased electricity demand. A more sophisticated design, figure 4.11b promises additional possibilities here, e.g. shifting the hysteresis limits depending on the available PV power. In this sense, in particular the large difference in CO_2 emissions compared to the optimisation results might be further reduced. On the other hand, the number of threshold values, i.e. selection of conditional nodes to be defined is increasing; additional investigations are required.
It should be noted that further consideration of the self sufficiency optimum probably plays a subordinate role in the current system configuration given the trade-off between total electricity price and emissions.

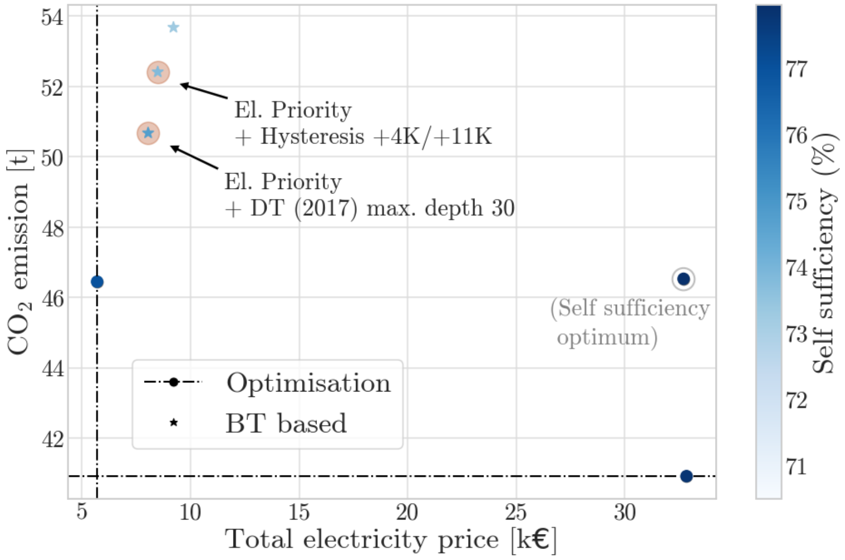


Figure 4.21: Comparison of exemplary BT runs with the optimisation for the full system. The shown results are obtained from insertion of certain heat system BT into the priority based realisation of the electricity system control.

Combination of the priority based electricity part BT and the hysteresis control +4K/+11K yields an annuity of 9.2 k€, emissions of 53.7 t and self sufficiency of 73.3% as shown in figure 4.21 for the year 2020. Better results are obtained here when introducing the safety DT instead. They are concluded together with the best runs for the sub systems in table. 4.5.

Overall favorable results are obtained using only the basic concepts of temperature based switching of the HP and component prioritization in the electricity part system.

Table 4.5: Overall results for the three different systems. The difference to the respective optimisation result is shown in the brackets. The results for the electricity system refer to $0.0 \leq SOC \leq 1.0$.

Best runs	**Self sufficiency**	**CO$_2$ emission [t]**	**Total price [k€]**
Heat system	0.0 (±0%)	32.9 (+11.4%)	25.1 (+4.6%)
El. System	83.6 (±0%)	27.2 (+17.2%)	-10.4 (±0%)
Full System	74.8 (–4.0%)	50.7 (+9.0%)	8.0 (+40.4%)

4.3 Exemplary Application Cases

It was shown, that at least for the proposed strategies and investigated data, no significant advantage or no advantage at all could be found for the use of DTs in combination with BTs. Also it has been mentioned, that BTs theirselves are not believed to have any advantage over DTs when it comes to regression, see also section 2.2.2. On the other hand, due to the resolution limitations (based on limitations of computational time) of the baseline optimisation reactivity advances of the BT were not exploited to any great extent. This is why, in this last chapter certain application cases should be investigated, where BTs are expected to be advantageous. The discussions are again limited to the two basic systems. Their activation frequency was increased to $1\,\mathrm{min}^{-1}$. In view of the simulation time for an entire year of around 12 min, this is considered a sensible choice.

4.3.1 Blocking of the Heat Pump

For the HP, switching behavior has already been discussed broadly. As especially frequent starting up of the compressor reduces the overall lifetime of a device [43, 53], it is favorable to define a minimum time difference before the next start up. This is in the range of several minutes, meaning that it only becomes relevant when the activation frequency is increased.

With the introduction of a time aware system, i.e. saving of a timestamp on the BB this can easily be included into a BT realisation for the heat system by adding a back propagated safety tree, see figure 4.22. If the blocking condition succeeds, the HP is turned off within this sub tree, otherwise this branch is skipped, leading to continuation of normal operation. Note that turn off behaviors within the tree need to be edited in order to write the timestamp of the last shut down onto the BB.

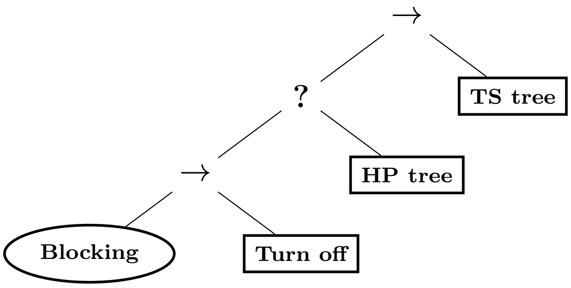


Figure 4.22: BT for the heating system with safety tree for blocking.

Such blocking is not only desired internally after deactivation of the heat pump but might also be added to a control system due to external blocking. An example are special tariffs, that allow for max. $\leq 2\,\mathrm{h}$ continuous, $\leq 6\,\mathrm{h}$ daily, and $\leq 960\,\mathrm{h}$ annual blocking (EVU blockage). A control system for such a scenario was created using the BT of figure 4.4 as a basis. The excess condition was set to an exceedance of 95% of the storage limit. A lower limit of the TES was set to the "predicted" demand for the subsequent two hours of maximum blocking ($E_{\mathrm{th}} = Q_{\mathrm{dem}} \times 2\,\mathrm{h}$). A respective external blocking pattern was created randomly under consideration of the above limits.

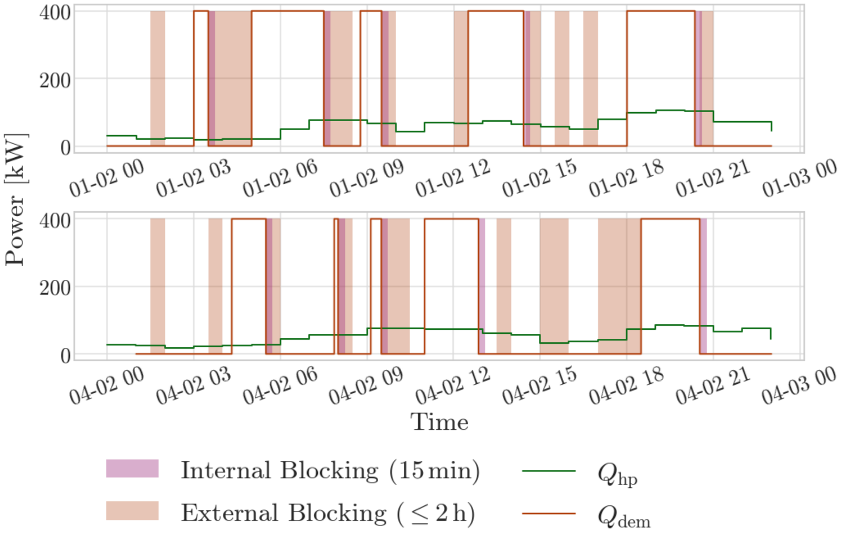


Figure 4.23: Operation of the heat system BT with blocking, figure 4.22 for two exemplary days. External blocking is randomly created, internal blocking determined by the turn off behavior of the tree.

In can be seen in figure 4.23, that this strategy allows for few overall start ups due to the possibly widespread temperature limits. For a test run over the course of a full year generated with `np.random.seed=2` for random blocking, the system was able to react to the blocking without deficits. Two exemplary days out of this are shown here, indicating both the blocking after shut down as well as external influence.

Of course this might not be the best strategy with respect to economical or ecological considerations, however combination with results from the previous section could be beneficial.

4.3.2 Peak Shaving and Storage Availability in the Electricity System

Finally, realisation of two functionalities, peak shaving and targeted storage provision, for the basic electricity system should be shown. The first can be particularly relevant for larger consumers. Here, avoidance of grid supply or feed-in load peaks, which possibly result in higher grid fees, is desired. Transferring this to the available data for a single household might nevertheless be an important use case, not last for grid relief [80]. Ultimately, it is just a scaling of the problem, whereby peak loads are usually less often scheduled in households.
Since peaks are mostly not observable in averaged demand data, such as those utilized in the previous investigations, 1 min data was applied within this section, see 3.1 and [64]. Irradiation data for deriving the PV power was again only available in hourly resolution. In correspondence with the order of magnitude of the demands, a fictitious system of 5.1 kWp solar panels and a battery with a capactiy of 8.0 kW h were simulated. Again this corresponds to approximately one half of the average daily demand. It will be seen that demand is temporarily up to three times higher than the peak load of the PV component.

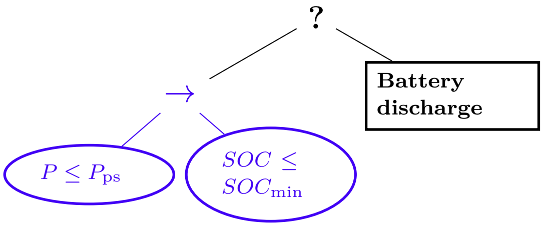


Figure 4.24: Sub tree of 4.10 with a modified battery condition.

A simple peak shaving mechanism is proposed using the basic structure of the BT in figure 4.10 with a modification of the battery condition according to figure 4.24. It is assumed that the main purpose of the battery consists in reducing the peak load and not necessarily in utilizing its entire charging range. Therefore the initial limits are set very narrow ($SOC_{\min} = 0.6$, $SOC_{\max} = 0.95$) and battery storage below a SOC of 0.6 is only used if a peak load occurs, $P > P_{\mathrm{ps}} = 2.5\,\mathrm{kW}$ (peak load threshold P_{ps}). This is conveniently implemented in a BT. An exemplary day, illustrating the behavior is shown in figure 4.25.

The effect on operation over an entire year is best visualized by using a histogram of the grid supply peaks, as shown in figure 4.26. Loads above the predefined threshold are significantly reduced, although due to the lack

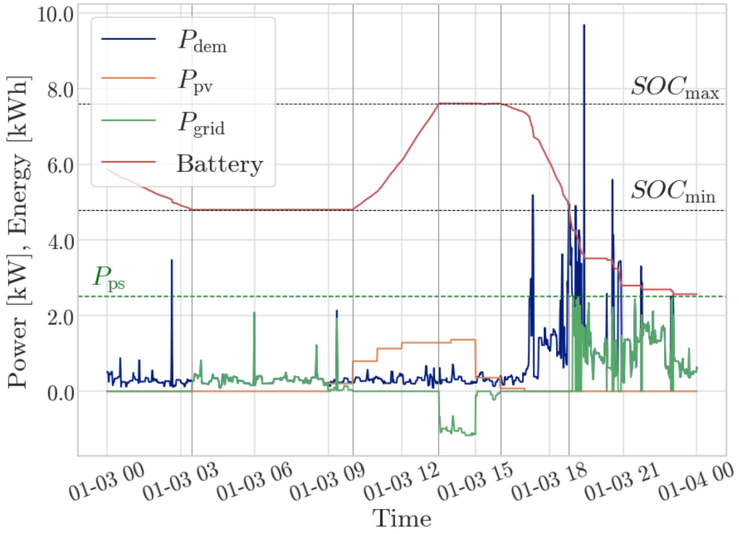


Figure 4.25: Exemplary day showing the operation of a BT with peak shaving mechanism. The lower limit of the battery *SOC* is undercut as soon as load peaks occur.

of predictability still present. Overall a reduction of 57.3% with respect to $0.2 \leq SOC \leq 0.8$ and 52.1% with respect to $0.2 \leq SOC \leq 0.95$ can be observed. This corresponds to total operation times of 78 h resp. 63 h.

A similar mechanism could be introduced for the reduction of feed-in peaks. However, this is not done here, partly because the PV peak power is already a means of limitation.

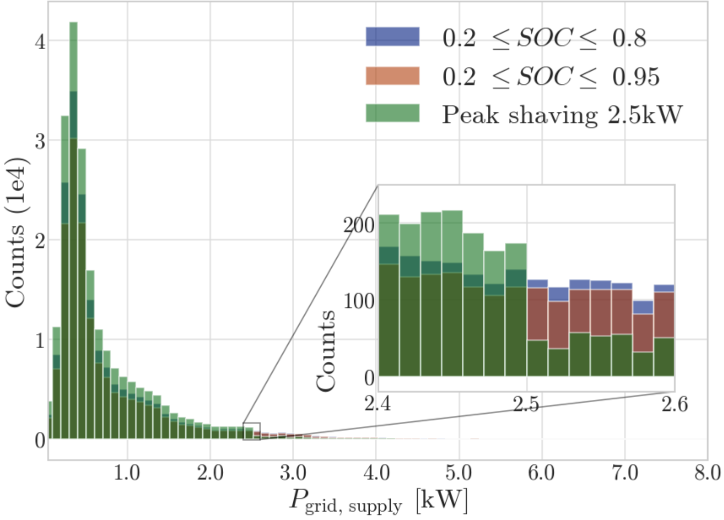


Figure 4.26: Effect of the peak shaving mechanism, shown in figure 4.24 for one year of operation.

To enhance this effect, the battery could be charged via the grid, e.g. before periods when a peak is expected. This was realised within the last exemplary use case. To provide a certain charge state of the battery at a certain time of the day, another BT realisation, based on the basic framework of section 4.1 is proposed in figure 4.27. Here the excess and charging conditions of the left sub tree are modified. Note that a similar condition would need to be input in the right part in order to avoid discharging at certain times.

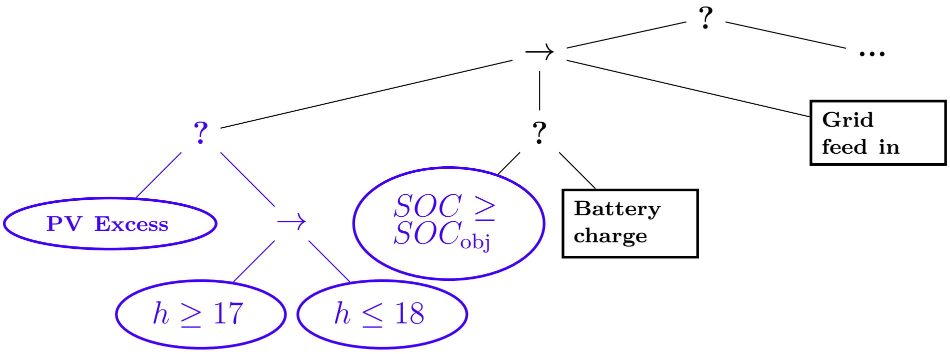


Figure 4.27: Sub tree of 4.10 with a modified (purple) excess and battery condition.

Charging and discharging is now not solely controlled by the exceedance of solar power over electricity demand, but also by the time of the day h. To secure a full battery at seven o clock in the evening, in the worst case, battery charging would needed to be started two hours before, assuming a C-rate of 0.5.

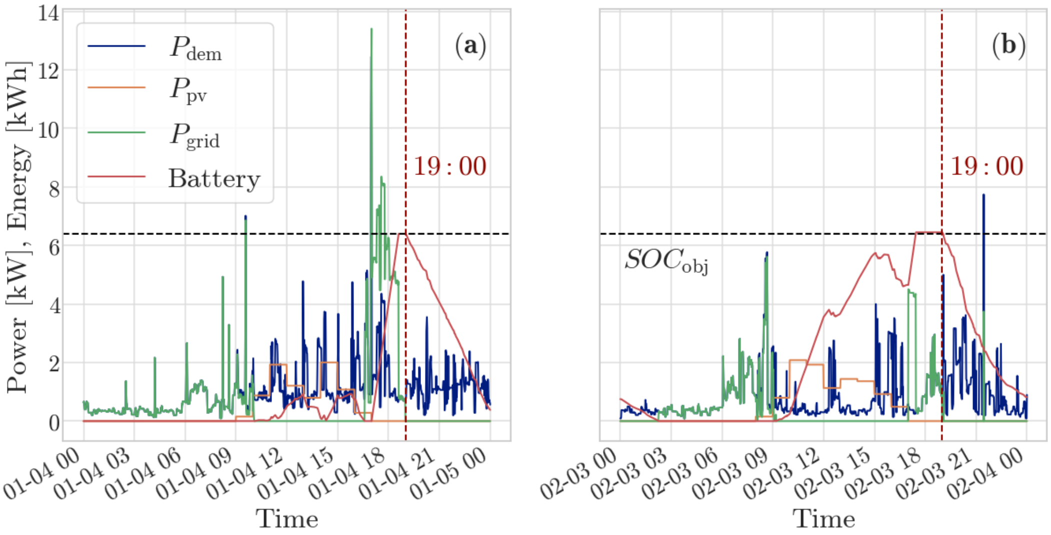


Figure 4.28: Two sample days illustrating the operation of the mechanism, shown in figure 4.27. The battery storage is forced to have a $SOC_{\mathrm{obj}} = 80\%$ at 19:00.

This is e.g. the case in figure 4.28a. Discharging is blocked while charging is supported by grid supply (if needed) as soon as the condition $h \geq 17$ is full filled. A second exemplary day showing the operation for less demand and

more avilable PV power can be seen in figure 4.28b.
Over the course of the year the battery is at an $SOC \geq 0.8$ in average at 17:10 and latest 18:36. The latter corresponds to 0% → 80% of charging with the full C-rate of 0.5, starting at 17:00. Further adaptions, e.g. connecting the start time to the current charge state would be sensible.

It can be seen, that in the particular examples, functionality requirements can affect each other, i.e. demand peaks occur upon charging of the battery. Ultimately control structures need to be created for the respective overall use case. This section was intended merely to give a brief insight into the possibilities of applying behavior trees to more specific problems.

5 Discussion: Potential of Behavior Trees in DES

Already in section 2.2 criteria were introduced for the evaluation of control structures, mainly of qualitative nature. In this context, the central aim was to illuminate the concept of BT as broadly as possible, yet always with a view to the comparative. Of course, the investigations in this work were subject to a number of limitations, which should be discussed first.

Limitations of this Work

Fundamentally, data availability first of all represents a limiting factor. Hourly annual data of 2017 and 2020 was utilized for the investigations, see 3.1. An extension of the data set to several years should be considered. Connected to this, the data resolution can be a crucial limitation. BT and FSM would most likely not operate with an activation duration of one hour in realistic applications. It is nevertheless assumed here, at least in section 4.2, for a "fair" comparison to the optimisations. Important objectives of the control structures, such as compliance with limit values for the storage levels, are thereby sometimes violated. Data resolution is not only subject to the available data, but also restricted by the computational time during optimization, as shown in section 4.2.2. A compromise must be found between meaningful component modelling for realistic results and the increased complexity of calculations, as can be observed in the heat pump model. Here, the implementation as a converter with constant COP would significantly reduce the computation time, whereby on the other hand changes in the partial load efficiency would be completely neglected.

Some limitations can also be identified for the conducted simulations. Especially the assumption of perfect operating conditions, neglecting the possibility of component failure or non accurate input (e.g. temperature measurements), undermines requirements for actual systems. All tests were confined to a singular ES dimensioning, moreover, component ageing and investment costs were omitted. Finally, it is important to recognize the constraints arising from the pre-selection of coding tools and their limited exploitation, influenced by the time constraints of the thesis project.

Comparison of the BT Concept

The previous chapter focused on two main aspects, development, i.e. development effort and comparison of the operation of different concepts.
Creating BT requires a certain change of mindset compared to the initially very intuitive FSM. In this hierarchy, the most ordinary approach is certainly that of DT using simple if ... else ... statements. When used as a stand-alone control system, the DT has the major disadvantage that all activities must be addressed in a single leaf node, reflecting the lack of feedback. These are therefore recommended rather as a superordinate decision-making tool.
However, all these structures possess the advantage of being graphically understandable. As mentioned by Olsson [34], visual editors are available in other domains, which are easier to utilise even without programming knowledge. Similar considerations may be appropriate in the energy building sector. It was nevertheless argued, that for FSM the actual development effort and maintainability (i.e. without such tools) exceeds that of BT, although the effect is reduced or even reversed with smaller structures. This is not a new insight, see section 2.2 [9, 32, 33, 34], but certainly needs further validation for the application in DES. One aspect that should be taken into consideration opposite control of ROS or non-player characters in the gaming industry is the expandability with regard to additional components. An entry point for this was outlined in section 4.1.4. How decisive this is in individual cases will vary from system to system. Not last, the results have shown that classic control without the need to introduce complex behavioral patterns can also be a sensible choice in basic ES.

The discussion of the operation is very much limited to the way it was simulated, as concluded in the previous section. No laboratory measurements were conducted here. Based on the assumption that BT and FSM can in principle be designed for the same output operation, only a comparison between the optimization and BT-based DT and classical concepts was performed.

In terms of computational effort, the latter promise to be much lighter controllers, directly affecting the needs in hardware. This is partly due to the fact, that no predictions are needed when compared to an optimisation based approach. However, this cannot necessarily be assumed to be decisive as algorithm capabilities and computational speed for MPC have increased immensely in recent times [31]. In the electricity system, for example, simulation speed for BT control is outperformed by the optimisation, although both achieve the same results in annuity. The more important fact is that the latter was based on an unrealistic, perfect foresight of demand. For this particular case, the use of a BT based control validated by an optimisation ultimately promises to be advantageous. Additional utilization of DT regressors without further adaptions in this context must be viewed cautiously. Although it is a well-developed instrument, DT tend to overfit. Selection of the best runs after simulation is oversimplified in the comparison and certainly, i.e. less meaningful in predicting applicability.

The use cases presented in the last part of the results chapter were not really compared to any of the other structures, rather they were meant to show some general capabilities of BT. Blocking of components and scheduled storage provision are two applications that have proven to be easy to implement. Furthermore they showed full functionality in the executed simulation runs. On the other hand the peak shaving mechanism showed, that further improvement and the establishing of more complex behavioral patterns is needed. The created implementation and simulation framework of BT opens up a wide range of experimental possibilities here. Comparison of the development effort of similar mechanisms, e.g. in FSM but also to capabilities of the optimisation tool used is a pending task.

Conclusion

Finally, the research questions formulated in chapter 1 shall be addressed. Behavior trees are generally not used as low-level controllers, but for superordinate task switching. In energy systems, this primarily involves managing heat and electricity flows by setting respective power values. The path from the synthesis of the ES under consideration to the creation of basic BT structures was discussed in detail. In this context, modularity proved to be the decisive advantage. Once derived, ground structures can be used to create a centralised control for several components, in the simplest case by adding them under a sequence node, but also by defining certain system states, e.g. excess and deficit of energy production. The ability to simply replace individual condition nodes with larger subtrees has significant potential for adding interpretable functionality. The advantage of reactivity could not really be proven for the shown ES and is not initially expected to play the same role as in production or robotics, for example.

In terms of operation concerning the minimization of KPI as total price or CO_2 emissions, there is no inherent advantage of BT. This is reflected in the fact that the implemented control mechanisms, e.g. hysteresis control, are of course not limited to a realisation with BT. However, the implementation of such mechanisms with BT could approach the model-based optimisations to a certain extent. In the considered heating system, some of the simulations carried out exceed the optimum price by just about 5%. Priority based switching in the electricity system just equals the optimisation. Emission-optimised operation is generally further off.

When comparing BT operation with an optimisation-based approach, its lower computational effort and the fact that no future data knowledge is required for stable operation certainly stand out as advantages. BT controllers should be easier to implement and consequently have lower hardware requirements. The main advantage over FSM is the expandability in terms of adding components and probably also functionality to a control structure. Operationally, depending on the implementations, differences are not to be expected here. Lastly DT are considered equally expressive, for BT no advantage can be determined in the decision-making process. However, when it comes to the subsequent execution of several actions, DT are quite unwieldy. As only a single leaf node is reached per activation, all actions would have to be defined there. The far better options for processing failures in BT illustrate this.

BT could be particularly useful for applications where a high degree of predictability and interpretability is required, which includes adaptation to the environment and the handling of errors (e.g. blocking, peak shaving) or for the fulfilment of certain recurring requirements (e.g. scheduled storage availability).

Ultimately, it should be stated that a sensible choice of concept always involves weighing up those various aspects. When referring back to the hierarchical structure of control in 2.1, it is most important to choose concepts in the right context. The use of BT and FSM for task switching should be emphasized and assessed with a view to the system size; DT or optimisation capabilities could rather take on the role of a high-level decision-making and planning.

6 Summary and Outlook

In this work, the concept of behavior trees, i.e. its transition from gaming and robotics to the energy sector, was examined with regard to various aspects. Existing theory, in particular for the comparison of the investigated structures BT, DT, FSM and to a lesser extent MPC, was reviewed and summarized. A framework for the operation of FSM and in particular BT was created to investigate the control opportunities in a heat system consisting of a heat pump and thermal storage, an electricity system consisting of a photovoltaic component, battery and grid supply as well as in a system combining those technologies.

It has been proven that both concepts basically enable the same operation in the heat and electricity systems. For their extension into control mechanisms for the complete system two strategies were proposed: component based and system state based combination of the ground structures. The first showed to be connected to a higher development effort for the FSM in terms of the applied measurement, cyclomatic complexity and graph edit distance as well as from a qualitative viewpoint. Estimates suggest an even greater effort for their system state based combination.

In the subsequent comparison of annual operation, total electricity price, CO_2 emissions and self-sufficiency were used as indicators, whereby the optimisations were carried out with oemof-solph to serve as a baseline. It showed that control with a hysteresis approach is competitive in the heat system with respect to the total price and surplus of only about 5% compared to the optimisation. In the electricity system prioritized operation of the components was best suited to approach the price optimum. Decision trees obtained by regression of 2017 optimal operation showed to be only partially beneficial. Reasonable operation in terms of annual KPI, again mainly the total price, in the complete system was shown exemplary for the component based combination of best running ground structures.

Exemplary functionalities using BT with higher activation frequency, where they are considered advantageous, were demonstrated and evaluated. A reliable blocking of the heat pump could be implemented here as well as a possibility to keep battery storage available at specific times. With the help of a straightforward peak shaving mechanism, load peaks in the analysed 1-minute data could be reduced by more than 50% compared to the previous normal operation. Finally an extensive discussion including review of the research questions was conducted in chapter 5.

Arising from this, there remain a number of challenges. The discussion is firstly lacking a realisation of the system state based FSM for the combined heat and electricity ES. As mentioned earlier, exploration of this type of extension, also for the BT promises a much more favorable utilization of the participating components.
The exemplary use cases presented in the last section require serious further development, not last because of their interdependence in the realisation of larger system. Here in particular, it would be interesting to further investigate the expansion possibilities of BT, a.o. to make wider use of their various node types or to evaluate losses in relation to certain KPIs against the establishment of new functionalities. As mentioned, those functionalities should also be implemented within the FSM concept to better understand the development effort and validate the benefits of using BT. Moreover, possible combination of advantage in the fusion of BT and FSM could be exploited.
Two opportunities for further research might be transferred to the context of DES: automated creation and exploration of stochastic BT [9]. A method for learning of BT using genetic programming algorithms is proposed by Iovino et al. [36]. The choice of instruments and the current implementation would definitely need to be refined in order to achieve a similar outcome, next to further difficulties as the definition of behaviors, and again selection of training data. However, comparison with current DT and optimisation results would be particularly attractive.
In the above investigations, a central control unit was used to monitor all system components. It is worth considering whether the implementation of different control mechanisms for individual components in a larger system would be more practicable, which leads to further questions. For example, how would the problem of synchronisation in the overall system, particularly with regard to grid stability, be solved if several components or systems share the same implementation? One solution to this question requires the consideration of stochastic BT. Analysing the operations of numerous actors within a system would again necessitate a significant expansion of the simulation framework.
Finally, as seen in [14], actual measurements in a laboratory environment using BT control are essential on the way to application in real systems.

Bibliography

[1] Umweltbundesamt. *Erneuerbare Energien in Deutschland. Daten zur Entwicklung im Jahr 2022.* `https://www.umweltbundesamt.de/sites/default/files/medien/1410/publikationen/2023-03-16_uba_hg_erneuerbareenergien_dt_bf.pdf`. Accessed: 2024-01-08 (cit. on pp. 1, 23).

[2] *Erneuerbare-Energien-Gesetz vom 21. Juli 2014, Änderung Dezember 2023.* `https://www.bundesregierung.de/breg-de/schwerpunkte/klimaschutz/novelle-eeg-gesetz-2023-2023972`. Accessed: 2024-01-08 (cit. on pp. 1, 49).

[3] Umweltbundesamt. *Projektionsbericht 2023 für Deutschland.* `https://www.umweltbundesamt.de/sites/default/files/medien/11850/publikationen/39_2023_cc_projektionsbericht_2023.pdf`. Accessed: 2024-01-08 (cit. on p. 1).

[4] Ting Wu, Dong-Ling Xu, and Jian-Bo Yang. "Decentralised energy and its performance assessment models." In: *Frontiers of Engineering Management* 8.2 (2021), p. 183. ISSN: 2095-7513. DOI: `10.1007/s42524-020-0148-7` (cit. on p. 1).

[5] Darioush Razmi and Tianguang Lu. "A literature review of the control challenges of distributed energy resources based on microgrids (MGs): past, present and future." In: *Energies* 15.13 (2022), p. 4676. DOI: `10.3390/en15134676` (cit. on pp. 1, 20).

[6] Antonius v. Perger, Philipp Gamper, and Rolf Witzmann. "Behavior trees for smart grid control." In: *IFAC-PapersOnLine* 55.9 (2022), p. 122. ISSN: 2405-8963. DOI: `10.1016/j.ifacol.2022.07.022` (cit. on pp. 1, 2).

[7] Michael Mateas and Andrew Stern. "A behavior language for story-based believable agents." In: *IEEE Intelligent Systems* 17.4 (2002), pp. 39–47. ISSN: 1541-1672. DOI: `10.1109/MIS.2002.1024751` (cit. on p. 2).

[8] Steven Rabin. *Game AI Pro: Collected Wisdom of Game AI Professionals*. Hoboken: CRC Press, 2013. ISBN: 978-1-4665-6597-5 (cit. on pp. 2, 9, 11).

[9] Michele Colledanchise and Petter Ögren. *Behavior Trees in Robotics and AI: An Introduction*. 2018. ISBN: 978-0-4294-8910-5. DOI: `10.1201/9780429489105` (cit. on pp. 2, 7–11, 14–16, 36, 68, 72).

[10] Martina Kajanova, Peter Bracinik, and Marek Roch. "Utilization of finite state machine approach for microgrid modeling." In: *Electrical Engineering* 102.1 (2020), pp. 53–63. ISSN: 0948-7921. DOI: `10.1007/s00202-019-00873-y` (cit. on p. 2).

[11] Alejandro Marzinotto et al. "Towards a unified behavior trees framework for robot control." In: *2014 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2014. DOI: `10.1109/icra.2014.6907656` (cit. on pp. 2, 5, 6, 10, 12).

[12] Kelleher R. Guerin et al. "A framework for end-user instruction of a robot assistant for manufacturing." In: *2015 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2015. DOI: `10.1109/icra.2015.7140065` (cit. on p. 2).

[13] Danying Hu et al. "Semi-autonomous simulated brain tumor ablation with ravenII surgical robot using behavior tree." In: *2015 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2015. DOI: `10.1109/icra.2015.7139738` (cit. on p. 2).

[14] Wang Jingsong. "Microgrid real-time decision control method based on behavior trees." In: *Proceedings of the 7th PURPLE MOUNTAIN FORUM on Smart Grid Protection and Control (PMF2022)*. Springer Nature Singapore, 2023, pp. 561–574. ISBN: 978-981-99-0063-3. DOI: `10.1007/978-981-99-0063-3_40` (cit. on pp. 2, 20, 72).

[15] Joseph J. DiStefano, Allen R. Stubberud, and Ivan J. Williams. *Feedback and Control Systems*. 2. ed. New York: McGraw-Hill, 2012. ISBN: 0-07-182948-2 (cit. on p. 5).

[16] Giuliano Donzellini et al. *Introduction to Digital Systems Design*. Cham: Springer International Publishing, 2019. ISBN: 978-3-319-92803-6. DOI: `10.1007/978-3-319-92804-3` (cit. on pp. 5, 37).

[17] Hai Lin and Panos J. Antsaklis. *Hybrid Dynamical Systems: Fundamentals and Methods*. Springer International Publishing, 2022. ISBN: 978-3-030-78731-8. DOI: `10.1007/978-3-030-78731-8` (cit. on p. 5).

[18] Lutz Priese and Katrin Erk. *Theoretische Informatik*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2018. ISBN: 978-3-662-57408-9. DOI: `10.1007/978-3-662-57409-6` (cit. on pp. 6–8, 14).

[19] Daniel Zelazo, Mehran Mesbahi, and Mohamed-Ali Belabbas. "Graph theory in systems and controls." In: *2018 IEEE Conference on Decision and Control (CDC)*. IEEE, 2018, pp. 6168–6179. DOI: `10.1109/cdc.2018.8619841` (cit. on p. 6).

[20] *python-statemachine documentation*. `https://python-statemachine.readthedocs.io/en/latest/index.html`. Accessed on 2023-12-21 (cit. on pp. 7, 30).

[21] Dingzhu Du and Ker-I Ko. *Theory of Computational Complexity*. 2. ed. Hoboken, NJ: Wiley, 2014. ISBN: 978-1-118-30608-6 (cit. on p. 8).

[22] Madan Somvanshi et al. "A review of machine learning techniques using decision tree and support vector machine." In: *2016 International Conference on Computing Communication Control and automation (ICCUBEA)*. IEEE, 2016, pp. 1–7. ISBN: 978-1-5090-3291-4. DOI: `10.1109/ICCUBEA.2016.7860040` (cit. on p. 8).

[23] Leo Breiman et al. *Classification And Regression Trees*. Routledge, 1984. ISBN: 9781315139470. DOI: `10.1201/9781315139470` (cit. on pp. 8, 9).

[24] Fabian Pedregosa et al. "Scikit-learn: machine learning in Python." In: *Journal of Machine Learning Research* 12 (2011), pp. 2825–2830. URL: `https://arxiv.org/abs/1201.0490` (cit. on pp. 8, 9, 30, 31).

[25] Michele Colledanchise and Petter Ögren. "How behavior trees modularize robustness and safety in hybrid systems." In: *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2014. DOI: `10.1109/iros.2014.6942752` (cit. on pp. 9, 16).

[26] Thomas Henn et al. "Verification of behavior trees using linear constrained horn clauses." In: *Formal Methods for Industrial Critical Systems*. Vol. 13487. Cham: Springer International Publishing, 2022, p. 211. ISBN: 978-3-031-15007-4. DOI: `10.1007/978-3-031-15008-1_14` (cit. on pp. 9, 16).

[27] Michele Colledanchise and Lorenzo Natale. "On the implementation of behavior trees in robotics." In: *IEEE Robotics and Automation Letters* 6.3 (2021), pp. 5929–5936. ISSN: 2377-3766. DOI: `10.1109/LRA.2021.3087442` (cit. on pp. 10, 28, 29).

[28] Michele Colledanchise, Ramviyas Parasuraman, and Petter Ögren. “Learning of behavior trees for autonomous agents.” In: *IEEE Transactions on Games* 11.2 (2019), pp. 183–189. ISSN: 2475-1502. DOI: `10.1109/TG.2018.2816806` (cit. on p. 11).

[29] Oliver Biggar, Mohammad Zamani, and Iman Shames. *On modularity in reactive control architectures, with an application to formal verification.* 2020. URL: `http://arxiv.org/pdf/2008.12515v3` (cit. on pp. 11, 14).

[30] Oliver Biggar, Mohammad Zamani, and Iman Shames. *A principled analysis of Behavior Trees and their generalisations.* 2020. URL: `http://arxiv.org/pdf/2008.11906v2` (cit. on pp. 11, 13, 15, 44).

[31] Sasa V. Raković. *Handbook of Model Predictive Control.* Cham: Birkhauser Verlag GmbH, 2019. ISBN: 978-3-319-77488-6. DOI: `10.1109/MCS.2020.3005257` (cit. on pp. 12, 20, 69).

[32] Edsger W. Dijkstra. “Letters to the editor: go to statement considered harmful.” In: *Communications of the ACM* 11.3 (1968), pp. 147–148. ISSN: 1557-7317. DOI: `10.1145/362929.362947` (cit. on pp. 14, 68).

[33] Matteo Iovino et al. *On the programming effort required to generate Behavior Trees and Finite State Machines for robotic applications.* 2022. URL: `http://arxiv.org/pdf/2209.07392v1` (cit. on pp. 14, 16, 28, 68).

[34] Magnus Olsson. *Behavior Trees for decision-making in Autonomous Driving.* Master thesis, KTH Computer Science and Communication. 2016. URL: `https://www.diva-portal.org/smash/get/diva2:907048/FULLTEXT01.pdf` (cit. on pp. 14, 16, 68).

[35] Thomas J. McCabe. “A complexity measure.” In: *IEEE Transactions on Software Engineering* SE-2.4 (1976), pp. 308–320. ISSN: 0098-5589. DOI: `10.1109/tse.1976.233837` (cit. on p. 14).

[36] Matteo Iovino et al. “Learning behavior trees with genetic programming in unpredictable environments.” In: *2021 IEEE International Conference on Robotics and Automation (ICRA).* IEEE, 2021. DOI: `10.1109/icra48506.2021.9562088` (cit. on pp. 14, 36, 48, 72).

[37] Zeina Abu-Aisheh et al. “An exact graph edit distance algorithm for solving pattern recognition problems.” In: *Proceedings of the International Conference on Pattern Recognition Applications and Methods.* SCITEPRESS - Science, 2015. DOI: `10.5220/0005209202710278` (cit. on p. 15).

[38] Oliver Biggar, Mohammad Zamani, and Iman Shames. *An expressiveness hierarchy of Behavior Trees and related architectures.* 2021. DOI: `10.48550/arXiv.2104.07919` (cit. on p. 15).

[39] John Estdale and Elli Georgiadou. "Applying the ISO/IEC 25010 quality models to software product." In: *Systems, Software and Services Process Improvement.* Vol. 896. Cham: Springer International Publishing, 2018, pp. 492–503. ISBN: 978-3-319-97924-3. DOI: `10.1007/978-3-319-97925-0_42` (cit. on p. 16).

[40] Maikhanh Dang. *Usability and maintainability of the software tools VIOLIN and CORAL.* Bachelor thesis, Duale Hochschule Baden-Württemberg. 2022. URL: `https://elib.dlr.de/189582/` (cit. on p. 16).

[41] D. Coleman et al. "Using metrics to evaluate software system maintainability." In: *Computer* 27.8 (1994), pp. 44–49. ISSN: 0018-9162. DOI: `10.1109/2.303623` (cit. on p. 16).

[42] Richard Zahoransky, ed. *Energietechnik: Systeme zur Energieumwandlung; Kompaktwissen für Studium und Beruf.* 6. ed. Wiesbaden: Springer Vieweg, 2013. ISBN: 978-3-8348-1869-0. DOI: `10.1007/978-3-658-34831-1` (cit. on pp. 17, 18).

[43] Holger Watter, ed. *Regenerative Energiesysteme.* Wiesbaden: Springer Fachmedien Wiesbaden, 2022. ISBN: 978-3-658-35867-9. DOI: `10.1007/978-3-658-35868-6` (cit. on pp. 17, 19, 20, 60).

[44] John A. Duffie, William A. Beckman, and Nathan Blair. *Solar Engineering of Thermal Processes, Photovoltaics and Wind.* Wiley, 2020. ISBN: 978-1-1195-4028-1. DOI: `10.1002/9781119540328` (cit. on p. 17).

[45] Guillermo Narsilio et al. "Geothermal energy: introducing an emerging technology." In: *Proceedings of the International Conference on Advances in Civil Engineering for Sustainable Development (ACESD 2014).* 2014, pp. 141–154. URL: `https://www.researchgate.net/publication/265643550` (cit. on p. 18).

[46] Chan O. Suong and Attakorn Asanakham. "Evaluation of a single stage heat pump performance by figure of merit (FOM)." In: *Energy Reports* 6 (2020), pp. 2735–2742. ISSN: 2352-4847. DOI: `10.1016/j.egyr.2020.09.038` (cit. on p. 18).

[47] Michael Sterner and Ingo Stadler. *Energiespeicher: Bedarf, Technologien, Integration.* Berlin and Heidelberg: Springer Vieweg, 2014. ISBN: 978-3-642-37379-4. DOI: `10.1007/978-3-642-37380-0` (cit. on pp. 18, 19).

[48] Aikaterini Chatzivasileiadi, Eleni Ampatzi, and Ian Knight. "Characteristics of electrical energy storage technologies and their applications in buildings." In: *Renewable and Sustainable Energy Reviews* 25 (2013), pp. 814–830. ISSN: 1364-0321. DOI: `10.1016/j.rser.2013.05.023` (cit. on p. 19).

[49] Seama Koohi-Fayegh and Marc A. Rosen. "A review of energy storage types, applications and recent developments." In: *Journal of Energy Storage* 27 (2020), p. 101047. ISSN: 2352-152X. DOI: `10.1016/j.est.2019.101047` (cit. on p. 19).

[50] Patrik Schönfeldt et al. "Simultaneous optimisation of temperature and energy in linear energy system models." In: *2022 Open Source Modelling and Simulation of Energy Systems (OSMSES)* (2022), pp. 1–6. DOI: `10.1109/OSMSES54027.2022.9768967` (cit. on pp. 19, 26, 27).

[51] Johannes Goeke. *Thermische Energiespeicher in der Gebäudetechnik.* Wiesbaden: Springer Fachmedien Wiesbaden, 2021. ISBN: 978-3-658-34509-9. DOI: `10.1007/978-3-658-34510-5` (cit. on pp. 19, 27).

[52] Duberney Murillo-Yarce et al. "A review of control techniques in photovoltaic systems." In: *Sustainability* 12.24 (2020), p. 10598. DOI: `10.3390/su122410598` (cit. on p. 20).

[53] Oliver Mercker, Peter Pärisch, and Gunter Rockendorf. "Taktverhalten von Sole-Wasser-Wärmepumpen–Messungen der thermischen Zeitkonstanten und ihre Bedeutung für die Jahresarbeitszahl." In: *Fifth German-Austrian IBPSA Conference, RWTH Aachen University.* 2014. URL: `www.ibpsa.org/proceedings/bausimPapers/2014/p1181_final.pdf` (cit. on pp. 20, 60).

[54] Eric O'Shaughnessy, Jesse R. Cruce, and Kaifeng Xu. "Too much of a good thing? Global trends in the curtailment of solar PV." In: *Solar Energy* 208 (2020), pp. 1068–1077. ISSN: 0038-092X. DOI: `10.1016/j.solener.2020.08.075` (cit. on p. 20).

[55] Rahmat Aazami et al. "Optimal Control of an Energy-Storage System in a Microgrid for Reducing Wind-Power Fluctuations." In: *Sustainability* 14.10 (2022), p. 6183. DOI: `10.3390/su14106183` (cit. on p. 20).

[56] Martina Capone and Elisa Guelpa. "Implementing optimal operation of multi-energy districts with thermal demand response." In: *Designs* 7.1 (2023), p. 11. DOI: `10.3390/designs7010011` (cit. on p. 20).

[57] Yongbao Chen et al. "Optimal control strategies for demand response in buildings under penetration of renewable energy." In: *Buildings* 12.3 (2022), p. 371. DOI: `10.3390/buildings12030371` (cit. on p. 20).

[58] Adrian Grimm et al. "Deduction of optimal control strategies for a sector-coupled district energy system." In: *Energies* 14.21 (2021), p. 7257. DOI: `10.3390/en14217257` (cit. on pp. 20, 21).

[59] Achintya Mukhopadhyay et al. *Dynamics and Control of Energy Systems.* Singapore: Springer Singapore, 2020. ISBN: 978-981-15-0535-5. DOI: `10.1007/978-981-15-0536-2` (cit. on p. 20).

[60] Steffen Wehkamp et al. "District energy systems: challenges and new tools for planning and evaluation." In: *Energies* 13.11 (2020), p. 2967. ISSN: 1996-1073. DOI: `10.3390/en13112967` (cit. on p. 21).

[61] Lucas Schmeling et al. "A generalised optimal design methodology for distributed energy systems." In: *Renewable Energy* 200 (2022), pp. 1223–1239. ISSN: 09601481. DOI: `10.1016/j.renene.2022.10.029` (cit. on pp. 21–24, 27).

[62] Harry Wirth. *Recent Facts about Photovoltaics in Germany.* `https://www.ise.fraunhofer.de/en/publications/studies/recent-facts-about-pv-in-germany.html`. Version 2023-09-27. Accessed on 2023-12-28 (cit. on p. 21).

[63] *Deutscher Wetterdienst.* `https://www.dwd.de/`. Accessed on 2023-12-29 (cit. on p. 22).

[64] Tjarko Tjaden et al. *Repräsentative elektrische Lastprofile für Einfamilienhäuser in Deutschland auf 1-sekündiger Datenbasis.* Datensatz. Licence: CC-BY-NC-4.0. 2019. URL: `https://solar.htw-berlin.de/wp-content/uploads/HTW-Repraesentative-elektrische-Lastprofile-fuer-Wohngebaeude.pdf` (visited on 11/20/2023) (cit. on pp. 23, 62).

[65] Fabian Backhaus. *Probabilistische Zeitreihenprognose für Gebäudelasten.* Master thesis, Carl von Ossietzky Universität Oldenburg. 2022 (cit. on p. 23).

[66] Samuel F. Fux. *Optimal energy management and component sizing of a stand-alone building energy system.* PhD thesis, ETH Zurich. 2013. DOI: `10.3929/ETHZ-A-009928622` (cit. on p. 24).

[67] Christian Inard and Jérôme Le Dréau, eds. *District Energy System Design: Simulation, Optimization and Decision Support*. Basel, Switzerland: MDPI - Multidisciplinary Digital Publishing Institute, 2020. ISBN: 978-3-03936-366-7. DOI: `20.500.12854/68898` (cit. on p. 24).

[68] *oemof.solph documentation*. `https://oemof-solph.readthedocs.io/en/stable/index.html`. Accessed on 2023-12-21 (cit. on pp. 25, 27, 31).

[69] Francesco Witte and Ilja Tuschy. "TESPy: Thermal Engineering Systems in Python." In: *Journal of Open Source Software* 5.49 (2020), p. 2178. DOI: `10.21105/joss.02178` (cit. on p. 26).

[70] Andreas Wagner. *Photovoltaik Engineering: Handbuch für Planung, Entwicklung und Anwendung*. 4. ed. Heidelberg et al.: Springer Vieweg, 2015. ISBN: 978-3-662-48639-9. DOI: `10.1007/978-3-662-58455-2` (cit. on p. 27).

[71] *py_trees documentation*. `https://py-trees.readthedocs.io/en/devel/index.html`. Accessed on 2023-12-21 (cit. on pp. 28–30, 52).

[72] *py_trees_ros documentation*. `https://docs.ros.org/en/kinetic/api/py_trees_ros/html/index.html`. Accessed on 2023-12-21 (cit. on p. 29).

[73] Simon Hilpert et al. "The open energy modelling framework (oemof) - A new approach to facilitate open science in energy system modelling." In: *Energy Strategy Reviews* 22 (2018), pp. 16–25. ISSN: 2211-467X. DOI: `10.1016/j.esr.2018.07.001` (cit. on p. 31).

[74] Janet Nagel. *Optimization of Energy Supply Systems: Modelling, Programming and Analysis*. Cham: Springer International Publishing, 2019. ISBN: 978-3-319-96354-9. DOI: `10.1007/978-3-319-96355-6` (cit. on p. 31).

[75] Uwe Krien et al. "oemof.solph — A model generator for linear and mixed-integer linear optimisation of energy systems." In: *Software Impacts* 6 (2020), p. 100028. ISSN: 2665-9638. DOI: `10.1016/j.simpa.2020.100028` (cit. on p. 31).

[76] Michael L. Bynum et al. *Pyomo — Optimization modeling in Python*. Vol. 67. Cham: Springer International Publishing, 2021. ISBN: 978-3-030-68927-8. DOI: `10.1007/978-3-030-68928-5` (cit. on p. 31).

[77] *mosaik documentation*. `https://mosaik.readthedocs.io/en/latest/index.html`. Accessed on 2023-12-28 (cit. on p. 33).

[78] Aric A. Hagberg, Daniel A. Schult, and Pieter J. Swart. "Exploring network structure, dynamics, and function using NetworkX." In: *Proceedings of the 7th Python in Science Conference*. OSTI Identifier: 960616. Pasadena, CA USA, 2008, pp. 11–15. URL: `https://www.osti.gov/biblio/960616` (cit. on p. 48).

[79] Jussi Vimpari. "Should energy efficiency subsidies be tied into housing prices?" In: *Environmental Research Letters* 16.6 (2021), p. 064027. ISSN: 1748-9326. DOI: `10.1088/1748-9326/abfeee` (cit. on p. 57).

[80] Julia Freier and Victor v. Loessl. "Dynamic electricity tariffs: designing reasonable pricing schemes for private households." In: *Energy Economics* 112 (2022), p. 106146. ISSN: 0140-9883. DOI: `10.1016/j.eneco.2022.106146` (cit. on p. 62).

Affirmation

I hereby affirm in lieu of oath that I have written this thesis independently and have not used any sources or aids other than those indicated. Furthermore, I affirm that I have followed the general principles of scientific work and publication as laid down in the guidelines of good scientific practice of the Carl von Ossietzky University Oldenburg.

Hiermit versichere ich an Eides statt, dass ich diese Arbeit selbstständig verfasst und keine anderen als die angegeben Quellen und Hilfsmittel benutzt habe. Außerdem versichere ich, dass ich die allgemeinen Prinzipien wissenschaftlicher Arbeit und Veröffentlichung, wie sie in den Leitlinien guter wissenschaftlicher Praxis der Carl von Ossietzky Universität Oldenburg festgelegt sind, befolgt habe.

Oldenburg, 28.01.2024

Place, Date Signature