energies

MDPI

Article

Reinforcement Learning Methods for the Stochastic Optimal Control of an Industrial Power-to-Heat System

Eric Pilling [1], Martin Bähr [2] and Ralf Wunderlich [1,*]

[1] Institute of Mathematics, Brandenburg University of Technology Cottbus-Senftenberg, 03013 Cottbus, Germany; eric.pilling@b-tu.de
[2] Department Simulation and Virtual Design, Institute of Low-Carbon Industrial Processes, German Aerospace Center, Weinbergstraße 10, 03050 Cottbus, Germany; martin.baehr@dlr.de
* Correspondence: ralf.wunderlich@b-tu.de

Abstract

The optimal control of sustainable energy supply systems, including renewable energies and energy storage, takes a central role in the decarbonization of industrial systems. However, the use of fluctuating renewable energies leads to fluctuations in energy generation and requires a suitable control strategy for the complex systems in order to ensure energy supply. In this paper, we consider an electrified power-to-heat system which is designed to supply heat in the form of superheated steam for industrial processes. The system consists of a high-temperature heat pump for heat supply, a wind turbine for power generation, a sensible thermal energy storage for storing excess heat, and a steam generator for providing steam. If the system's energy demand cannot be covered by electricity from the wind turbine, additional electricity must be purchased from the power grid. For this system, we investigate the cost-optimal operation, aiming to minimize the electricity cost from the grid by a suitable system control depending on the available wind power and the amount of stored thermal energy. This is a decision-making problem under uncertainty regarding the future prices for electricity from the grid and the future generation of wind power. The resulting stochastic optimal control problem is treated as finite-horizon Markov decision process for a multi-dimensional controlled state process. We first consider the classical backward recursion technique for solving the associated dynamic programming equation for the value function and compute the optimal decision rule. Since that approach suffers from the curse of dimensionality, we also apply reinforcement learning techniques, namely Q-learning, that are able to provide a good approximate solution to the optimization problem within reasonable time.

Keywords: stochastic optimal control; Markov decision process; dynamic programming; Q-learning; power-to-heat system; renewable energy; cost-optimal energy management

check for updates

Academic Editor: Constantinos S. Psomopoulos

Received: 18 December 2025
Revised: 5 February 2026
Accepted: 8 February 2026
Published: 17 February 2026

Copyright: © 2026 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license.

1. Introduction

Nowadays, the supply of process heat for industrial processes by conventional systems leads to high CO_2 emissions, as these are predominantly based on the combustion of fossil fuels. The electrification of heat generation through the use of novel technologies, such as high-temperature heat pumps (HTHP), is a potential measure to reduce these emissions. In combination with renewable energy sources, the sustainable heat supply for industrial processes is based on complex systems that require realistic modeling as well as cost- and emission-optimized system operation. In particular, electrified energy supply systems

face the challenge of determining a cost-optimal operating strategy due to the fluctuating power generation from renewable energies, while ensuring the required heat demand of the industrial process is satisfied. In this context, a potential industrial power-to-heat (P2H) system with an HTHP providing heat for a steam generator (SG), see Figure 1, was recently proposed by Walden et al. [1]. This P2H system uses the availability of an on-site wind turbine (WT) to generate its own electricity to power the HTHP. This reduces the cost of purchasing electricity from the power grid. These costs can be further reduced by using a thermal energy storage (TES), which serves to balance out the fluctuating generation of renewable energy. The overproduction of electricity can then be stored as thermal energy and used later to supply the system with its own resources.

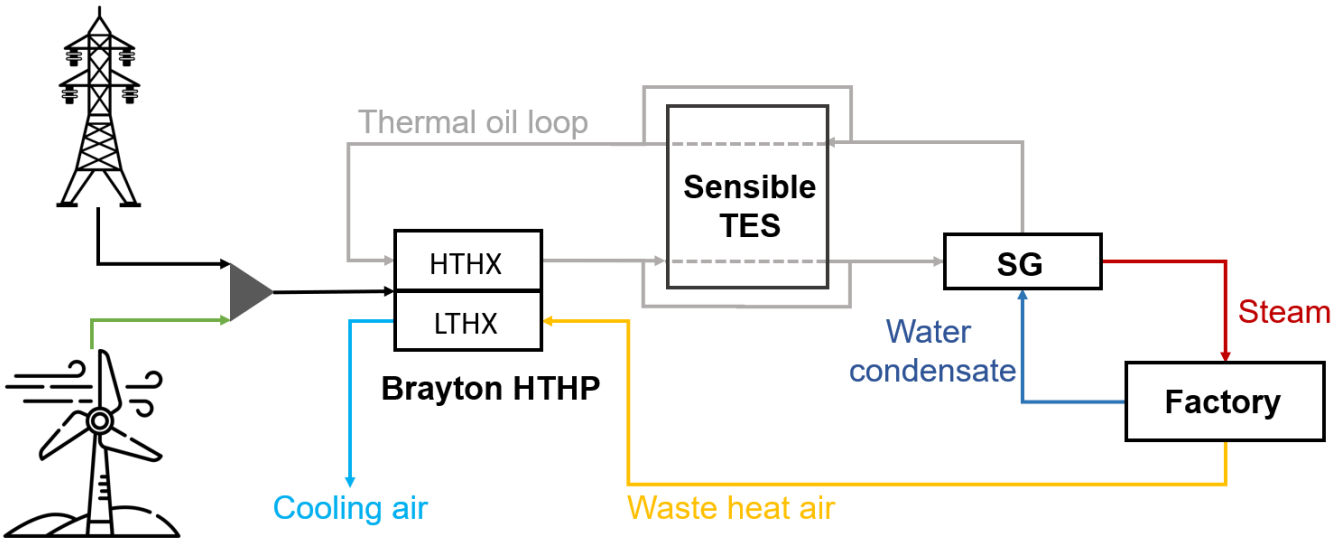


Figure 1. Illustration of the investigated industrial P2H system for electrified steam generation proposed in [1]. The thermal system consists of an HTHP, a TES, and a SG, which are connected via a thermal oil loop. The HTHP uses a waste heat air stream as the heat source and is powered by electricity from a WT or the power grid in order to provide constant heat supply.

In [1], the cost-optimal operation of this electrified system with the aim of minimizing the total cost of grid power was treated as a deterministic optimization problem and solved using methods of algebraic nonconvex, nonlinear programming theory. In addition, it was assumed that future wind power generation and electricity prices were already known in advance. However, in real-world scenarios, the problem of the optimal management and operation of such systems is a decision-making problem under uncertainty, as precise forecasts of future wind energy supply and electricity prices are not possible. Therefore, the problem must be formulated in a stochastic framework.

In this context, the following typical question needs to be addressed: At what time and at what rate should energy be stored in or withdrawn from the TES to reduce the costs of grid electricity? To address this question, we will treat the cost-optimal management of the underlying industrial P2H system as a stochastic optimal control problem and solve it using Markov decision process (MDP) theory.

Literature Review on Optimal Management of Industrial P2H Systems.

This literature is embedded in numerous studies on optimization problems for energy systems, particularly for electrical and thermal microgrids. However, many of the articles only briefly describe the underlying model and methods. The optimization problems are mainly related to technical aspects, and the control problem is solved with commercial

optimization software. The mathematical aspects of the optimal control of energy systems are generally not sufficiently addressed.

The optimal management of *combined heating and power systems* has recently been studied by several authors. In Testi et al. [2], the optimal integration of electrically powered heat pumps into a hybrid decentralized energy supply system is investigated. The authors propose a multi-criteria stochastic optimization method to determine the integrated optimal dimensioning and operation of energy supply systems under uncertainties regarding climate, land use, energy demand, and fuel costs. Kuang et al. [3] study the optimization of the off-design operation of combined heating and power systems, including energy storage. The cost-optimal control of a heating system for residential buildings with geothermal storage under uncertainties regarding heat supply and demand as well as energy prices is investigated in Takam et al. [4]. A review on the optimal energy management of combined cooling, heating, and power microgrids is given in Gu et al. [5]. Further contributions on combined heating, cooling, and power system can be found in [6,7] and the references therein.

In recent years, *machine learning methods* have also been increasingly used to solve optimization problems. For example, Bui et al. [8] and Mohammed et al. [9] model a microgrid energy management system with battery storage that is connected to the energy grid and distributed energy sources. To obtain an optimal operation strategy that aims to handle loads, prices, and the decision of charging or discharging the battery, Q-learning [10] is used. Nakabi and Toivanen [11] considered a similar application, but they used and compared different state-of-the-art reinforcement learning algorithms like Q-learning, deep deterministic policy gradients, or proximal policy optimization to achieve the optimal management of their microgrid model. Another application of MDPs is proposed by Yu et al. [12], who apply reinforcement learning to a home energy management system. In addition to the usage of a battery storage and the connection to the power grid as well as renewable energies, the household utilizes a heating, ventilation, and air conditioning system that needs to be operated as cost efficiently as possible. Belloni et al. [13] use a MDP formulation of a system with a WT and battery storage to obtain the optimal control with dynamic programming. Thermal storage devices combined with a heat pump are used in the papers of Ridder et al. [14] and Chenzi et al. [15], which use dynamic programming and Q-learning, respectively.

Literature Review on Stochastic Optimal Control.

The cost-optimal management of energy supply systems under uncertainty can be treated mathematically as a stochastic optimal control problem. There is extensive literature on this theory. Much of this literature focuses on solution methods based on the dynamic programming principle. For continuous-time problems in which the controlled state process is a diffusion or jump-diffusion process, this results in the Hamilton–Jacobi–Bellman equation, which acts as a necessary optimality condition; we refer to Fleming and Soner [16], Pham [17], and Oksendal and Sulem [18]. This is a nonlinear partial differential equation that can usually only be solved by *numerical methods*, as in Shardin and Wunderlich [19] and Chen and Forsyth [20].

For solutions to discrete-time problems, *Markov decision process* theory offers an algorithm based on backward recursion. Standard references are Bäuerle and Rieder [21], Puterman [22], Hernández-Lerma and Lasserre [23], and Powell [24]. We note that such MDPs can be also obtained from a time discretization of continuous-time control problems.

For high-dimensional state spaces, the solution of MDPs suffers from the curse of dimensionality. To overcome this problem, powerful numerical methods have been developed in recent years. Examples are the least squares Monte Carlo method introduced in [25,26], approximate dynamic programming, Q-learning and related reinforcement learning methods [10,24,27], and optimal quantization methods [28] as well as neural network [29,30] and deep learning methods [31,32].

Our Contribution.

In this article we present a mathematical model for the operation of an industrial P2H system with an HTHP and a TES. It explicitly takes into account the stochasticity of intermittent renewable energy sources such as wind power and the fluctuating market prices for electricity in the power grid. These variables are modeled by suitable stochastic processes that are calibrated to real-world data. Furthermore, the model takes into account that permanent changes in the operating points should be avoided. We therefore treat the cost-optimal energy management problem a stochastic optimal control problem in discrete-time, in which the controls are kept constant between two discrete points in time. To avoid unnecessary time discretization errors, the dynamics of the state and system variables that reflect the operation of the system are still treated in continuous time.

The optimization problem is formulated as an MDP and solved using dynamic programming methods. Since the state of the control problem is three-dimensional, the numerical solution already faces the curse of dimensionality. The problem becomes even more serious when we extend and refine our stylized model to include more details of the HTHP operation. Then the computational effort for the numerical solutions becomes prohibitively high. Therefore, in this paper, we investigate reinforcement learning methods such as Q-learning to find a faster numerical approximate solution. Finally, we present the results of extensive numerical experiments in which we compare the results of the different numerical methods.

Paper Organization.

Section 2 is devoted to a thorough mathematical modeling of the considered industrial P2H system. It introduces the state and control variables and additional system variables, as well as the underlying assumptions on the P2H system. In Section 3, a MDP formulation of the stochastic optimal control problem is derived. This section provides details on the formulation of the stochastic processes for wind speed and the electricity price, state and control constraints, and the cost functions of the optimal control problem. In Section 4, the classical approach to solve MDP problems by backward recursion based on the associated Bellman equation is presented. Approximate solutions based on reinforcement learning techniques, particularly Q-learning, are described in Section 5. Finally, Section 6 presents results of numerical experiments in which the optimal control problem is solved using the methods proposed in Sections 4 and 5. The Supplementary Material collects proofs and technical results that have been removed from the main text to avoid interrupting the reader with these details. The flowchart in Figure 2 provides an overview of the workflow of the paper. It summarizes and illustrates the interplay of the content presented in Sections 2–5.

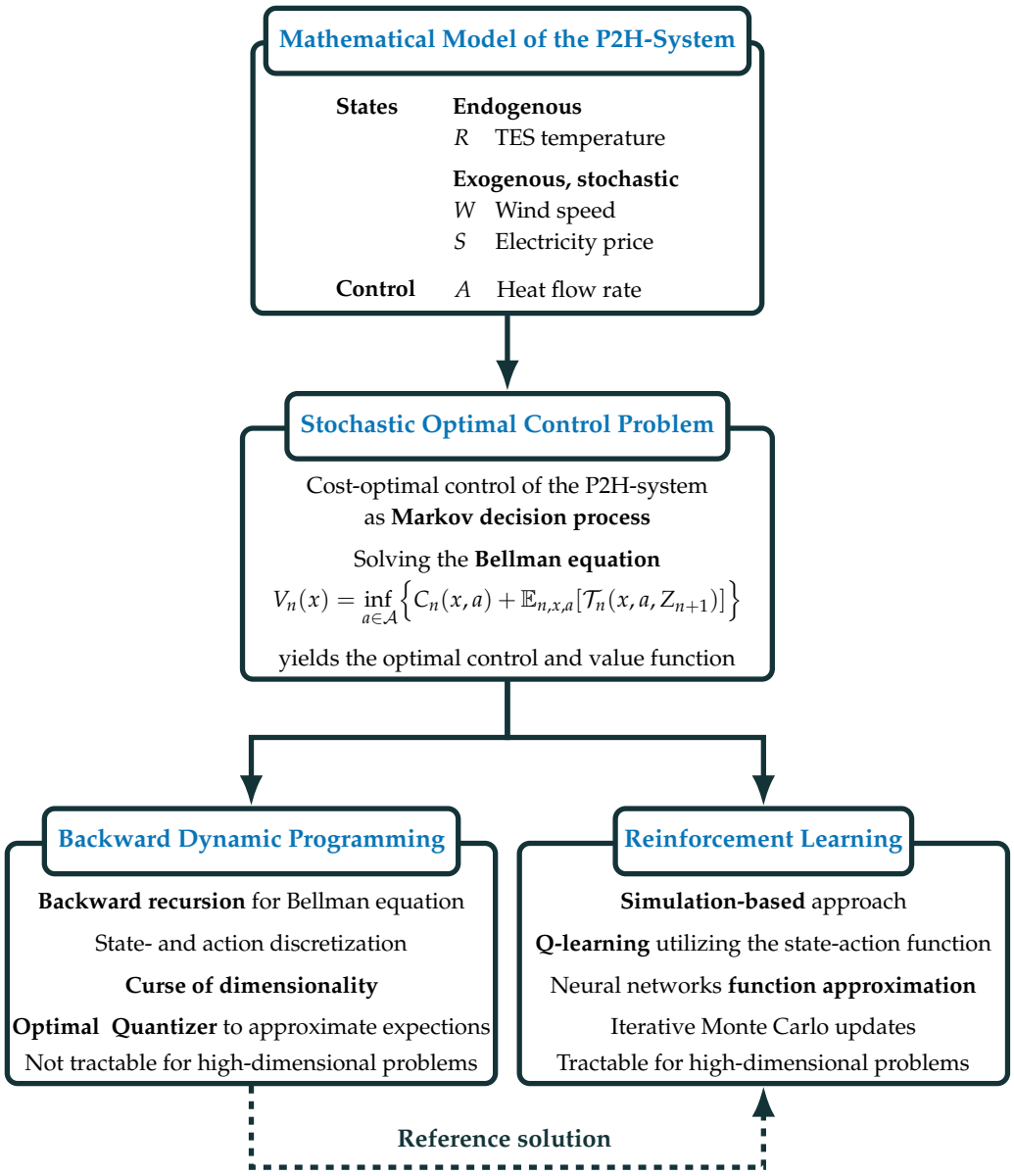


Figure 2. Flowchart representing the workflow and structure of this paper. Each block contains key facts about the associated section. The titles of the boxes are hyperlinks and can be used to navigate to the corresponding section in the article.

2. Mathematical Modeling of the Industrial P2H System

In this section, a mathematical model for the operation of the industrial P2H system is developed. It is treated as a control system with an endogenous state variable that can be influenced by a control variable and exogenous stochastic states. The P2H system is subject to various operational constraints that lead to state and control constraints. For more technical details about the underlying P2H system, we refer the reader to [1]. In the following, we first briefly introduce the industrial P2H system and then describe its mathematical modeling as a control system.

2.1. Industrial P2H System

The industrial P2H system based on renewable energy, heat pump, and energy storage shown in Figure 1 is designed to supply constant process heat in the form of superheated steam at 13 bar and 215 °C. The system consists of the following four components:

(i) An on-site *wind turbine* that generates renewable electricity;

(ii) A *high-temperature heat pump* for heat supply, which is powered by electricity from the WT or the power grid;
(iii) A concrete-based sensible *thermal energy storage* to store excess energy in times of high wind power production or low grid electricity prices;
(iv) A *steam generator* to provide constant process steam.

The thermal system components (ii)–(iv) are connected via a thermal oil loop. Note that the TES is specifically designed for short periods of time, i.e., for storing excess heat for some hours or a few days.

A more detailed system configuration is depicted in Figure 3. The HTHP generates high-temperature process heat up to 350 °C with a thermal output of up to around 2.2 MW, which is fed into a thermal fluid loop with thermal oil as the heat transfer fluid (HTF). Via a fluid bypass, the charging factor $l^{\mathrm{C}} \in [0,1]$ determines the proportion of the HTF that is routed through the TES, while the remaining proportion $1 - l^{\mathrm{C}}$ passes directly into the SG. The bypass is used to regulate the charging process of the TES. During charging, the hot HTF flows through the cooler TES and heats the storage medium (concrete).

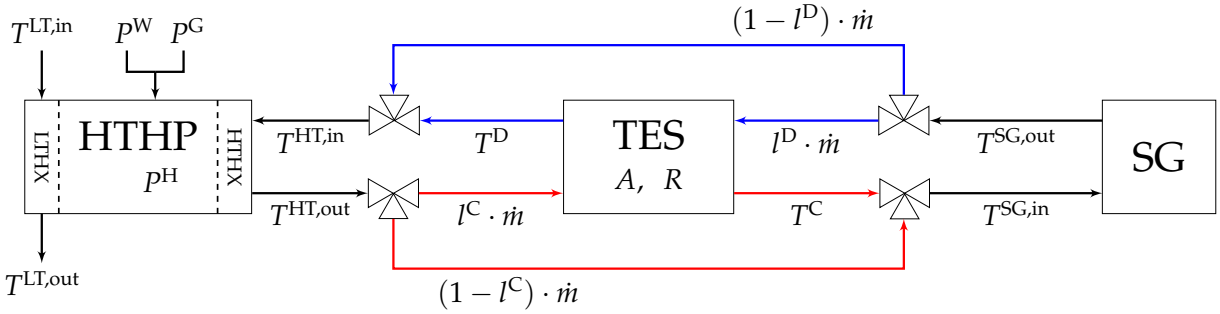


Figure 3. Detailed flow diagram [1] of the studied industrial P2H system (cf. Figure 1) with HTHP, TES, and SG. The charging and discharging factors $l^{\mathrm{C}}, l^{\mathrm{D}} \in [0,1]$ determine the heat flow to the SG and the HTHP depending on the thermal state of the TES. Exemplarily, the *red* lines indicate the charging mode if $l^{\mathrm{C}} \in (0,1)$, and the *blue* lines represent the discharging mode for $l^{\mathrm{D}} \in (0,1)$. Simultaneous charging and discharging is not allowed. *Charging* mode is characterized by $l^{\mathrm{C}} \in (0,1]$, $l^{\mathrm{D}} = 0$, *discharging* by $l^{\mathrm{D}} \in (0,1]$, $l^{\mathrm{C}} = 0$, and *idle* mode by $l^{\mathrm{C}} = l^{\mathrm{D}} = 0$. For better understanding, the following HTHP and SG operating temperatures are used: $T^{\mathrm{HT,out}} \in [239, 350]$, $T^{\mathrm{HT,in}} \in [177, 250]$, $T^{\mathrm{SG,in}} \in [239, 324]$, and $T^{\mathrm{SG,out}} \in [183, 193]$.

A second bypass from the SG outlet returning to the high-temperature heat exchanger (HTHX) of the HTHP is used to discharge the TES. The discharge factor $l^{\mathrm{D}} \in [0,1]$ specifies the proportion of the HTF that is passed through the TES, such that the remaining part $1 - l^{\mathrm{D}}$ enters directly into the HTHP. During the discharging process, the HTF cooled in the SG flows through the warmer TES and lowers the temperature of the storage medium. As the fluid now enters the HTHX at a higher temperature, the HTHP's electricity consumption is reduced.

In idle operation, characterized by $l^{\mathrm{C}} = l^{\mathrm{D}} = 0$, the TES is completely bypassed by the HTF. At the low-temperature heat exchanger (LTHX) of the HTHP, a waste heat air stream at 60–100 °C from the industrial consumer is used as heat source. We note that the cold air outlet stream at the LTHX is not used for cooling applications in the current configuration. Further, it is assumed that the system components, particularly the HTHP and SG, operate in steady state. This means that the dynamic behavior of the components during operating point changes is neglected.

In our setup, we use the heat flow rates that determine the charging and discharging operations of the TES to describe the control of the P2H system. These rates have a direct functional relationship with the HTHX inlet and outlet temperatures, the HTHP's

compressor shaft speed, and electricity consumption, which is described in more detail below in Section 2.4. The HTHP electricity consumption that is not covered by wind energy determines the amount of electricity drawn from the grid, resulting in a direct functional relationship with the running electricity costs that are included in the performance criterion of the optimization problem. More details follow below in Section 3.4.

The mathematical description of nonlinear component models for HTHP and SG is based on process simulation software that also takes into account the part-load behavior of the heat pump. Based on this, the physical characteristics are then approximated by algebraic surrogate models that appropriately mimic the input-output behavior of the components. For our purposes, it is sufficient to model the state of charge of the TES only by its spatially averaged temperature and neglect the detailed spatial temperature distribution, as, for example, in [33,34]. This avoids complex calculation of internal heat propagation and facilitates the solution of the optimization problem. This simplification preserves the dynamics of the TES while significantly reducing the computational complexity, making it suitable for integration into energy management optimization.

The actual WT power output is typically modeled by a function of the wind speed at rotor height. This dependence is given by the so-called power curve, which is explained in Section S6.3 of the Supplementary Material. We emphasize that the system and component modeling does not take into account pressure and heat losses as well as friction losses and also no electricity consumption of auxiliary systems such as fluid pumps. For example, fluid oil pumps are not considered because the main power consumption is based on the HTHP, which is many times larger than the pumps, so the total power consumption only shows a small error.

The design and dimensioning of the system components was determined by engineering calculations, for which we refer to [1]. Recall, the aim is to determine the cost-optimal operation of the P2H system that minimizes the expected total costs over a finite planning horizon $t_E > 0$ from the purchase of grid electricity and the revenues from the sale of WT overproduction, taking into account the uncertainty of the fluctuating wind energy supply and electricity prices. To derive the mathematical formulation of this optimization problem in form of an MDP in Section 3, we describe in the following the details of the state and control variables, additional system variables, and operational constraints.

2.2. Time Discretization

While the state and system variables of the P2H system evolve continuously over time, the control variables available to the controller are typically not changed permanently but only at discrete points in time and are then kept constant until the next time point. This is caused by the fact that operating the HTHP, i.e., changing the HTHX outlet temperature by varying the compressor shaft speed, induces thermal stresses in the heat exchangers during transient operations. For this reason, rapid changes in the operating points should be avoided and limited to a few discrete points in time.

We therefore divide the planning horizon $[0, t_E]$ into $N \in \mathbb{N}$ uniformly spaced subintervals of length $\Delta t = t_E/N$ and define the time grid points $t_n = n\Delta t$ for $n = 0, \ldots, N$. Let $G : [0, t_E] \to \mathbb{R}$ be a given continuous-time function. In this work we use the short-hand notation $G_n := G(t_n)$ for the sampled value at the time grid point t_n. The control variables and some related system variables are assumed to be piecewise constant functions on the time grid introduced above and take the values $G(t) = G_n$ for $t \in [t_n, t_{n+1})$ with $n = 0, \ldots, N-1$. The remaining system variables and particularly the state variables of the control problem are treated as continuous-time functions governed by certain equations that capture the dynamics of the system. However, the controller only uses the values at the time grid points for the control decisions.

The electricity price for trading on the intraday spot market is not quoted continuously over time, but generally only every 15 min. For the sake of simplicity, we model this price as a continuous-time stochastic process and discretize it accordingly.

2.3. State and Control Variables

State Variables.

For the formulation of the stochastic optimal control problem, the following three variables describe the state of the control system at time $t \in [0, t_E]$:

$R(t)$, the average TES temperature [°C];
$W(t)$, the wind speed [m/s];
$S(t)$, the electricity price [€/MWh].

These state variables can be divided into endogenous and exogenous quantities. Here, R is the only *endogenous* variable that is subject to the control action, while W and S are exogenous variables and determined outside the model. The storage temperature R changes during charging or discharging operation and is directly related to the associated heat flow rate, which forms the control variable introduced below.

The *exogenous* states, on the other hand, are stochastic variables, as the wind speed and the electricity price are subject to a certain degree of uncertainty, meaning that future values are not known exactly in advance and are afflicted with considerable forecasting errors. They must therefore be modeled as stochastic processes, where the detailed description is deferred to Section 3.1.

Control Variable.

In our model, we suppose that the operation of a P2H system at time $t \in [0, t_E]$ is controlled by

$A(t)$, the heat flow rate related to the TES [kW]

with values in some action or control space $\mathcal{A} \in \mathbb{R}$ which will be specified below. In the following, we use the sign convention that a positive heat flow rate corresponds to charging, while negative values of A indicate discharging. The idle mode is represented by $A(t) = 0$. We show in Section 2.4 that specifying this variable is sufficient to adjust the other system variables describing the HTHP operation accordingly. There, we will also explain how the HTHP electricity demand depends on the control A, which in turn determines the demand for electrical energy drawn from the grid when this demand is not fully covered by wind energy, or the overproduction of wind energy that can be fed into the grid. The operational costs associated with the control A are derived in Section 3.4. Note that the controller's choice of the heat flow rate A is subject to various constraints, which we explain in detail in Section 3.3.

Endogenous State Variable.

The control A directly determines the dynamics of the only controlled/endogenous state variable in our control system, namely, TES temperature R. It results from an energy balance that describes the change in thermal energy in the TES due to the inflow and outflow of energy during charging and discharging. Since the TES is designed to store excess heat for short periods of hours or a few days, we neglect heat losses to the environment. Then the change in thermal energy in the time interval $[t_a, t_b]$ with $0 \leq t_a < t_b \leq t_E$ in the continuous-time setting is given by $\int_{t_a}^{t_b} A(s)\,\mathrm{d}s$. On the other hand, it is equal to $m_s c_{p,s}(R(t_b) - R(t_a))$, where $c_{p,s}$ and m_s denote the specific heat capacity (assumed to be temperature independent) and the mass of the storage medium, respectively. The values

used are specified in Supplementary Material, Section S2. This leads to the following relation for the TES temperature for $t \in [t_a, t_b]$

$$R(t) = R(t_a) + \frac{1}{m_s c_{p,s}} \int_{t_a}^{t_b} A(s)\, \mathrm{d}s. \tag{1}$$

As already mentioned in Section 2.2, we make the following

Assumption 1 (Piecewise constant control)**.** *Control A is kept constant between two consecutive grid points of the time discretization, i.e.,*

$$A(t) = A(t_n) =: A_n, \quad t \in [t_n, t_{n+1}),\ n = 0, \ldots, N-1. \tag{2}$$

Below, in Section 2.4, we notice that the assumption of constant heat flow rates in the TES corresponds to constant oil temperatures at the inlet and outlet of the HTHX within the periods between the time grid points. This avoids rapid changes in the HTHP operation. Only at the time grid points t_n the heat flow rate changes immediately from A_{n-1} to A_n, whereby the transient behavior of the HTHP and TES components is neglected.

Under the above assumption that the heat flow rate A is piecewise constant and does not vary within a time period $[t_n, t_{n+1})$, the dynamics (1) of the TES temperature within such a period simplifies then to

$$R(t) = R_n + \frac{1}{m_s c_{p,s}} A_n (t - t_n), \quad \text{and particularly} \quad R_{n+1} = R_n + \frac{1}{m_s c_{p,s}} A_n \Delta t, \tag{3}$$

where we recall the notation $R_n = R(t_n)$ and $\Delta t = t_{n+1} - t_n$.

2.4. Additional System Variables and Operational Constraints

The mathematical modeling of the operation of the P2H system, as shown in Figures 1 and 3, requires the consideration of several additional variables that have not been included in the set of state and control variables. They are referred to as system variables and are subject to certain operational constraints, which are explained in this subsection. These system variables and their dynamics are needed to derive state-dependent control constraints of the control problem that we formulate in Section 3. However, the system variables are regarded as internal variables whose specific values do not need to be observed by the controller and which are not included in the decision-making process. The latter is based solely on knowledge of the state variables.

To formulate the mathematical model discussed in this work, we make the following simplification.

Assumption 2 (Constant HTF mass flow and waste heat temperature)**.** *The mass flow $\dot{m}$ of the thermal oil stream and the temperature $T^{LT,in}$ of the waste heat air stream at the LTHX inlet are constant over the entire period $[0, t_E]$.*

Note that in [1] the HTF mass flow rate $\dot{m}$ can vary within a certain range; here, we assume that $\dot{m}$ is constant. Although this simplification leads to a lower system flexibility, it avoids the introduction of an additional control variable and thus reduces the complexity of the problem as well as the computational effort required to compute the numerical solution.

2.4.1. Steam Generator

The constant heat demand of the SG must be satisfied at all times, leading to the following relations between the mass flow $\dot{m}$ and temperatures at the inlet $T^{\mathrm{SG,in}}$ and outlet $T^{\mathrm{SG,out}}$ of the SG:

$$T^{\mathrm{SG,in}} = F^{\mathrm{SG,in}}(\dot{m}) \quad \text{and} \quad T^{\mathrm{SG,out}} = F^{\mathrm{SG,out}}(\dot{m}). \tag{4}$$

The nonlinear functions $F^{\mathrm{SG,in}}$ and $F^{\mathrm{SG,out}}$ represent surrogate models for the underlying energy balances and are generated using process simulations, see Section S1 of the Supplementary Material. According to Assumption 2, the mass flow $\dot{m}$ is constant in our model, so $T^{\mathrm{SG,in}}$ and $T^{\mathrm{SG,out}}$ are also constant over the entire period $[0, t_{\mathrm{E}}]$. It is obvious that $T^{\mathrm{SG,in}} > T^{\mathrm{SG,out}}$ because the SG can simply be understood as a heat exchanger used to supply the factory with superheated steam.

2.4.2. High-Temperature Heat Pump

We now describe the relationship between the HTHP, particularly the inlet and outlet temperatures at the HTHX and its electricity demand, and the control variable.

Piecewise Constant HTHX Inlet and Outlet Temperature.

According to Assumption 1, the control variable A representing the heat flow rate in the TES is piecewise constant, i.e., $A(t) = A_n$ in each interval $[t_n, t_{n+1})$ for $n = 0, \ldots, N-1$. This property is transferred to the HTHX inlet and outlet temperatures $T^{\mathrm{HT,in}}$ and $T^{\mathrm{HT,out}}$, which result from the given system configuration in which the TES is integrated, the fact that $T^{\mathrm{SG,in}}$ and $T^{\mathrm{SG,out}}$ are constant on $[0, t_{\mathrm{E}}]$ and the following energy balances.

In period $[t_n, t_{n+1})$ during charging, we have an inflow of thermal energy from the HTHP to the thermal oil loop with rate $n_{\mathrm{H}} \dot{m} c_{\mathrm{p,f}} T^{\mathrm{HT,out}}(t)$ and an outflow from this loop to the SG with constant rate $n_{\mathrm{H}} \dot{m} c_{\mathrm{p,f}} T^{\mathrm{SG,in}}$. Here, $c_{\mathrm{p,f}}$ denotes the specific heat capacity (assumed to be temperature independent) of the thermal oil and n_{H} the number of HTHPs operating in parallel in the underlying P2H system, which is not explicitly shown in Figure 3). The values used are specified in Supplementary Material, Section S2. Since we neglect losses to the environment, the difference between the two rates gives the constant inflow rate A_n to the TES during the charging process. Recall that $A_n > 0$ holds in charging mode, thus it follows $T^{\mathrm{HT,out}} > T^{\mathrm{SG,in}}$.

Analogously, during discharging, there is an inflow of thermal energy from the SG with constant rate $n_{\mathrm{H}} \dot{m} c_{\mathrm{p,f}} T^{\mathrm{SG,out}}$ and an outflow to the HTHP with rate $n_{\mathrm{H}} \dot{m} c_{\mathrm{p,f}} T^{\mathrm{HT,in}}(t)$. The difference of the two rates determines the outflow rate from the TES, which is A_n. Since $A_n < 0$ during discharging, it follows $T^{\mathrm{HT,in}} > T^{\mathrm{SG,out}}$.

Based on this, we obtain the relations

$$\begin{aligned} A_n &= n_{\mathrm{H}} \dot{m}\, c_{\mathrm{p,f}} (T^{\mathrm{HT,out}}(t) - T^{\mathrm{SG,in}}) > 0, \quad \text{during charging and} \\ A_n &= n_{\mathrm{H}} \dot{m}\, c_{\mathrm{p,f}} (T^{\mathrm{SG,out}} - T^{\mathrm{HT,in}}(t)) < 0, \quad \text{during discharging} \end{aligned} \tag{5}$$

from which follows for the HTHX outlet and inlet temperatures

$$\begin{aligned} T^{\mathrm{HT,out}}(t) &= T_n^{\mathrm{HT,out}} = \tau^{\mathrm{out}}(A_n) \quad \text{with} \quad \tau^{\mathrm{out}}(a) = T^{\mathrm{SG,in}} + \tfrac{a^+}{n_{\mathrm{H}} \dot{m}\, c_{\mathrm{p,f}}}, \quad \text{and} \\ T^{\mathrm{HT,in}}(t) &= T_n^{\mathrm{HT,in}} = \tau^{\mathrm{in}}(A_n) \quad \text{with} \quad \tau^{\mathrm{in}}(a) = T^{\mathrm{SG,out}} + \tfrac{a^-}{n_{\mathrm{H}} \dot{m}\, c_{\mathrm{p,f}}}, \end{aligned} \tag{6}$$

which are constant in each of the N time periods. Here, $a^{\pm} = \max(\pm a, 0)$ denotes the positive and negative parts of a. From this, the following unified notation can be derived for the mapping given in (5), which reads

$$A_n = g^{\mathrm{HF}}(T_n^{\mathrm{HT,in}}, T_n^{\mathrm{HT,out}}), \tag{7}$$

with $g^{\mathrm{HF}}(\tau^{\mathrm{in}}, \tau^{\mathrm{out}}) = n_{\mathrm{H}} \dot{m}\, c_{\mathrm{p,f}}(\tau^{\mathrm{out}} - \tau^{\mathrm{in}} - (T^{\mathrm{SG,in}} - T^{\mathrm{SG,out}}))$. Note that $T_n^{\mathrm{HT,in}} = T^{\mathrm{SG,out}}$ during charging, $T_n^{\mathrm{HT,out}} = T^{\mathrm{SG,in}}$ during discharging, and both equations hold true during idle periods.

HTHX Outlet Temperature.

From [1] it is known that there is a complex relationship between the HTHX oil outlet temperature $T^{\mathrm{HT,out}}$ and the HTHX oil inlet temperature $T^{\mathrm{HT,in}}$, the HTF mass flow $\dot{m}$, the waste heat air temperature $T^{\mathrm{LT,in}}$ at the LTHX inlet, and the compressor shaft speed D, which can be expressed by a surrogate model in terms of a multivariate cubic polynomial F_1 as $T^{\mathrm{HT,out}}(t) = F_1(T^{\mathrm{HT,in}}(t), \dot{m}, T^{\mathrm{LT,in}}, D(t))$. As we know from (6) that $T^{\mathrm{HT,in}}$ and $T^{\mathrm{HT,out}}$ are stepwise functions, i.e., $T^{\mathrm{HT,out}}(t) = T_n^{\mathrm{HT,out}}$ and $T^{\mathrm{HT,in}}(t) = T_n^{\mathrm{HT,in}}$ in each interval $[t_n, t_{n+1})$ for $n = 0, \ldots, N-1$, and based on the Assumption 2, it follows directly that D is also constant between two subsequent discrete time points, so we obtain

$$T_n^{\mathrm{HT,out}} = F_1(T_n^{\mathrm{HT,in}}, \dot{m}, T^{\mathrm{LT,in}}, D_n). \tag{8}$$

More specifically, the compressor shaft speed D can be varied (only at discrete time grid points) to determine the pressure ratio within the HTHP and thus the thermal oil temperature $T^{\mathrm{HT,out}}$ at the HTHX outlet. Details on the definition of the polynomial function F_1 are provided in Section S1 of the Supplementary Material.

HTHP Electricity Consumption.

Another complex relationship from [1] holds for the electricity consumption P^{H} of the HTHP. Again, this is expressed by a surrogate model in the form of a multivariate quadratic polynomial F_2 for $n = 0, \ldots, N$ as

$$P_n^{\mathrm{H}} = n_{\mathrm{H}} F_2(T_n^{\mathrm{HT,in}}, \dot{m}, T^{\mathrm{LT,in}}, D_n). \tag{9}$$

The factor n_{H} in (9) as already mentioned above, denoting the number of HTHPs operating in parallel, takes into account that the surrogate model represents only a single HTHP. The detailed definition of F_2 can be found in Section S1 of the Supplementary Material.

Relation (9) together with Assumption 2 and the piecewise constant quantities $T_n^{\mathrm{HT,in}}$ as well as D_n implies that P_n^{H} is also constant between two subsequent discrete time points. This demand has to be covered by the sum of the power P^{G} drawn from or fed into the grid and the power P^{W} generated by the WT, both of which are generally time-varying, meaning that

$$P_n^{\mathrm{H}} = P^{\mathrm{G}}(t) + P^{\mathrm{W}}(t), \quad t \in [t_n, t_{n+1}),\ n = 0, \ldots, N-1. \tag{10}$$

For $P^{\mathrm{G}} > 0$, electricity is drawn (purchased) from the grid, for $P^{\mathrm{G}} < 0$, electricity is fed (sold) into the grid. Note that the dependence of the WT power P^{W} on the wind speed W is described by the so-called power curve, $P^{\mathrm{W}} = P_{\mathrm{WT}}(W)$. This is a nonlinear function that grows cubically [35] at medium wind speeds until it reaches the rated power. This value is kept constant for higher speeds and is set to zero above the cut-out speed and below a cut-in speed of the turbine. More details are given in Section S6.3 of the Supplementary Material.

Dependence of Electricity Consumption on the Control.

Based on (8) and (9), the HTHP's electricity consumption P_n^{H} can be determined at each time point t_n for a given control A, which determines, by (6), the temperatures $T_n^{\mathrm{HT,out}}$ and $T_n^{\mathrm{HT,in}}$ of the HTF at the outlet and inlet of the HTHX, respectively. Suppose that at time t_n the control is set to be $A_n = a$, then (6) implies $T_n^{\mathrm{HT,out}} = \tau^{\mathrm{out}}(a)$ and $T_n^{\mathrm{HT,in}} = \tau^{\mathrm{in}}(a)$, and recall that $\dot{m}$ and $T^{\mathrm{LT,in}}$ are constants. Then, in a first step, the corresponding shaft speed $D_n = d$ is determined by solving (8) for the unknown d, i.e., $\tau^{\mathrm{out}}(a) = F_1(d) := F_1(\tau^{\mathrm{in}}(a), \dot{m}, T^{\mathrm{LT,in}}, d)$. Since F_1 is a cubic polynomial in d, the solution is among the real-valued roots of this polynomial. For the given F_1 and using the fact that the shaft speed is restricted to values within the interval $[d_{\min}, d_{\max}]$, defined by technical conditions [1], there is a unique root $d^* = d^*(a)$ in this interval. In a second step, the corresponding electricity consumption P_n^{H} is obtained by substituting d^* into (9) via

$$P_n^{\mathrm{H}} = n_{\mathrm{H}} F_2(\tau^{\mathrm{in}}(a), \dot{m}, T^{\mathrm{LT,in}}, d^*(a)). \tag{11}$$

To emphasize the dependence of P^{H} on the control A, we introduce the function π^{H} that maps A to the electricity consumption for $n = 0, \ldots, N-1$, i.e.,

$$P_n^{\mathrm{H}} = \pi^{\mathrm{H}}(A_n), \quad \text{with} \quad \pi^{\mathrm{H}}(a) = n_{\mathrm{H}} F_2(\tau^{\mathrm{in}}(a), \dot{m}, T^{\mathrm{LT,in}}, d^*(a)). \tag{12}$$

2.4.3. Thermal Energy Storage Operational Modes

The TES operation can be divided into three operational modes as follows: charging, discharging and idle, where we assume that simultaneous charging and discharging is not allowed. From the previous subsections, the following relations are known for each operational mode:

$$\begin{array}{lllll} \text{Charging:} & A_n > 0, & T^{\mathrm{HT,in}} = T^{\mathrm{SG,out}}, & l^{\mathrm{C}} \in (0,1], \\ \text{Discharging:} & A_n < 0, & T_n^{\mathrm{HT,out}} = T^{\mathrm{SG,in}}, & l^{\mathrm{D}} \in (0,1], \\ \text{Idle:} & A_n = 0, & l^{\mathrm{C}} = l^{\mathrm{D}} = 0. & \end{array} \tag{13}$$

In charging mode, the excess thermal energy is stored in the sensible TES and increases its medium temperature, while during discharging the cooler HTF with temperature $T^{\mathrm{SG,out}}$ absorbs heat from the TES and decreases the temperature of the storage medium. For details on the charging and discharging mode, we refer to Section S3 of the Supplementary Material. In idle mode, the HTF bypasses the TES completely and obviously implies $T_n^{\mathrm{HT,out}} = T^{\mathrm{SG,in}}$ and $T_n^{\mathrm{HT,in}} = T^{\mathrm{SG,out}}$, which also follows directly from Equations (S3.2) and (S.3.5) given in the Supplementary Material. There, we provide further details, including the dependence of the time-varying charging and discharging factors $l^{\mathrm{C}}, l^{\mathrm{D}}$ on the TES temperature R and the chosen control A_n.

3. Stochastic Optimal Control Problem

In this section, we formulate the stochastic optimization problem using the Markov decision process framework for the cost-optimal energy management of the industrial P2H system introduced above. Most of the MDP theory can be found in the books of Bäuerle and Rieder [21], Hernandez and Lerma [23], Powell [24], and Puterman [22]. We would like to refer the interested reader to those books for further information about MDPs. The goal is to find the optimal control that minimizes the expected total cost for electricity consumption from the grid over a finite planning horizon, taking into account the uncertainties of future wind energy and electricity prices as well as the revenues from selling overproduction. The derived stochastic control problem consists of the following blocks: state dynamics,

state and control constraints, operational cost functions, state and control space, transition operator, and performance criterion.

3.1. State Dynamics

In Section 2.3, we introduced the three state variables R, W, and S. The dynamics of the endogenous (or controlled) variable R representing the TES temperature is already given in (1) and (3). Here, we focus on the two exogenous states, the wind speed W and the electricity price S. Starting with the continuous-time approach, we derive recursions for the state values at the discrete time points t_n for $n = 0, \dots, N$. We decompose W and S into a non-random function that captures the seasonal patterns and a stochastic Ornstein–Uhlenbeck process that is mean-reverting to zero to describe the unpredictable fluctuations.

Throughout this paper, all stochastic processes and random variables are supposed to be defined on a filtered probability space $(\Omega, \mathcal{F}, \mathbb{F}, \mathbb{P})$. In particular, that space carries a two-dimensional Brownian motion $B = (B^W, B^S)$, with two independent standard Brownian motions B^W, B^S on $[0, t_E]$, which will be used below to drive stochastic differential equations (SDEs) describing the dynamics of W and S. The filtration $\mathbb{F}$ is assumed to be generated by B, that is, $\mathbb{F} = \mathbb{F}^B = (\mathcal{F}^B(t))_{t\in[0,t_E]}$ with the σ-algebras $\mathcal{F}^B(t) = \sigma\{B(s), s \le t\}$, augmented by the $\mathbb{P}$-nullsets, so that $\mathbb{F}$ satisfies the usual assumptions. While, the market price of electricity S can also take negative values, the wind speed W is always non-negative. We therefore replace W with $\log W$ and assume for $t \in [0, t_E]$

$$\begin{aligned} \log W(t) &= \mu_W(t) + Y^W(t), \\ S(t) &= \mu_S(t) + Y^S(t). \end{aligned} \tag{14}$$

Here, the functions $\mu_W, \mu_S : [0, t_E] \to \mathbb{R}$ describe seasonal patterns, and Y^W and Y^S are Ornstein–Uhlenbeck processes defined by SDEs

$$\begin{aligned} \mathrm{d}Y^W(t) &= -\lambda_W Y^W(t)\,\mathrm{d}t + \sigma_W\,\mathrm{d}B^W(t), \\ \mathrm{d}Y^S(t) &= -\lambda_S(c_W Y^W(t) + Y^S(t))\,\mathrm{d}t + \sigma_S\,\mathrm{d}B^S(t), \end{aligned} \tag{15}$$

with mean reversion speeds $\lambda_W, \lambda_S > 0$, diffusion coefficients $\sigma_W, \sigma_S > 0$, and a constant $c_W \ge 0$. Due to the different natures of wind speed and electricity price, we assume that $\lambda_W \ne \lambda_S$ to simplify our analysis. A positive constant c_W leads to a negative correlation of the wind speed W and the price process S, see Lemma 1 and Proposition 1 below. This is often observed in energy markets.

The two SDEs above have analytical solutions that allow us to derive the following closed-form expressions for the joint distribution of the pair of random variables W and S as follows, with the proofs provided in Section S5 of the Supplementary Material.

Lemma 1 (Distribution of solutions Y^W, Y^S to the SDEs (15))**.** *Let $0 \le t_a < t_b \le t_E$. Then the solutions of the SDEs (15) on $[t_a, t_b]$ to given initial values $Y^W(t_a) = y_W$ and $Y^S(t_a) = y_S$ with $y_W, y_S \in \mathbb{R}$ are Ornstein–Uhlenbeck processes with*

$$\begin{aligned} Y^W(t) &= y_W e^{-\lambda_W \tau} + \int_{t_a}^{t} \sigma_W e^{-\lambda_W(t-r)}\,\mathrm{d}B^W(r), \\ Y^S(t) &= y_S e^{-\lambda_S \tau} - \lambda_S c_W \int_{t_a}^{t} e^{-\lambda_S(t-r)} Y^W(r)\mathrm{d}r + \int_{t_a}^{t} \sigma_S e^{-\lambda_S(t-r)}\mathrm{d}B^S(r), \end{aligned} \tag{16}$$

for $t \in [t_a, t_b]$ and $\tau = t - t_a$. The conditional distribution of the pair $(Y^W(t), Y^S(t))$ given $(Y^W(t_a), Y^S(t_a)) = (y_W, y_S)$ is bivariate Gaussian with mean $m_Y(\tau, y_W, y_S)$ and positive definite covariance matrix $\Sigma_Y(\tau)$ given by

$$m_Y(\tau, y_W, y_S) = \begin{pmatrix} m_{Y^W}(\tau, y_W) \\ m_{Y^S}(\tau, y_W, y_S) \end{pmatrix}, \quad \Sigma_Y(\tau) = \begin{pmatrix} \Sigma^2_W(\tau) & \Sigma_{WS}(\tau) \\ \Sigma_{WS}(\tau) & \Sigma^2_S(\tau) \end{pmatrix}, \tag{17}$$

where for $\tau \geq 0$

$$\begin{aligned} m_{Y^W}(\tau, y_W) &= y_W e^{-\lambda_W \tau}, \\ m_{Y^S}(\tau, y_W, y_S) &= y_S e^{-\lambda_S \tau} - \frac{\lambda_S c_W}{\lambda_S - \lambda_W} y_W \left(e^{-\lambda_W \tau} - e^{-\lambda_S \tau}\right), \\ \Sigma^2_W(\tau) &= \tfrac{\sigma^2_W}{2\lambda_W}\left(1 - e^{-2\lambda_W \tau}\right), \\ \Sigma^2_S(\tau) &= \Sigma^2_{Y^S}(\tau) + \tfrac{(\lambda_S c_W)^2}{(\lambda_S - \lambda_W)^2}\left[\Sigma^2_W(\tau) + \tfrac{\sigma^2_W}{\sigma^2_S}\Sigma^2_{Y^S}(\tau) - \tfrac{2\sigma^2_W}{\lambda_S + \lambda_W}\left(1 - e^{-(\lambda_S + \lambda_W)\tau}\right)\right], \end{aligned} \tag{18}$$

with $\Sigma^2_{Y^S}(\tau) = \frac{\sigma^2_S}{2\lambda_S}\left(1 - e^{-2\lambda_S \tau}\right)$, and the covariance

$$\Sigma_{WS}(\tau) = -\tfrac{\lambda_S c_W}{\lambda_S - \lambda_W}\left[\Sigma^2_W(\tau) - \tfrac{\sigma^2_W}{\lambda_S + \lambda_W}\left(1 - e^{-(\lambda_S + \lambda_W)\tau}\right)\right]. \tag{19}$$

It holds that $\Sigma_{WS}(\tau) \leq 0$ for $c_W \geq 0$ with equality for $c_W = 0$.

Combining the above result for $t_a = t_n$ and $t_b = t_{n+1} = t_a + \Delta t$ with the definitions from (14), and recalling the notation $W_n = W(t_n), S_n = S(t_n)$ for $n = 0, \ldots, N$, we obtain the following result for the joint distribution of $(\log W_{n+1}, S_{n+1})$ given $(\log W_n, S_n)$. This will be useful for the construction of the transition operator and the corresponding transition kernel for the MDP's state process below in (34).

Proposition 1 (Conditional distribution of $(\log W_{n+1}, S_{n+1})$ given $(\log W_n, S_n)$)**.** *The conditional distribution of the pair $(\log W_{n+1}, S_{n+1})$ given $(\log W_n, S_n) = (\log w, s)$ with $w > 0, s \in \mathbb{R}$ is bivariate Gaussian with mean*

$$m^{WS}_{n+1}(w, s) = \begin{pmatrix} m^W_{n+1}(w) \\ m^S_{n+1}(w, s) \end{pmatrix}, \tag{20}$$

and the constant and positive definite covariance matrix $\Sigma = \Sigma_Y(\Delta t)$ and

$$\begin{aligned} m^W_{n+1}(w) &= \mu_W(t_{n+1}) + m_{Y^W}(\Delta t, \log w - \mu_W(t_n)), \\ m^S_{n+1}(w, s) &= \mu_S(t_{n+1}) + m_{Y^S}(\Delta t, \log w - \mu_W(t_n), s - \mu_S(t_n)), \end{aligned} \tag{21}$$

where m_{Y^W}, m_{Y^S} and Σ_Y are given in Lemma 1.

Note that the wind speed W follows a log-normal distribution because $\log W$ is Gaussian. The above result on the conditional distribution of the pairs $(\log W_n, S_n)$ and the fact that the dynamics of the stochastic fluctuations Y^W, Y^S are driven by Brownian motions, i.e., processes with independent increments, can be used to derive a recursion for the discrete-time dynamics of the sequence of these pairs which is driven by a sequence of independent standard normally distributed random vectors.

Corollary 1 (Discrete-time dynamics of wind speed and energy price)**.** *Let the Cholesky decomposition of the symmetric and positive definite covariance matrix* Σ *given in Proposition 1 be of the form*

$$\Sigma = AA^\top \quad with \quad A = \begin{pmatrix} \Sigma_W & 0 \\ \rho\Sigma_S & \sqrt{1-\rho^2}\Sigma_S \end{pmatrix} \quad and \quad \rho = \frac{\Sigma_{WS}}{\Sigma_W \Sigma_S}, \tag{22}$$

where ρ *denotes the associated correlation coefficient. Then there exists a sequence* $(Z_n)_{n=1,\dots,N}$ *of independent standard normally distributed random vectors* $Z_n = (Z_n^W, Z_n^S)^\top \sim \mathcal{N}(0_2, I_2)$ *such that*

$$(\log W_{n+1}, S_{n+1}) = m_{WS}(n+1, W_n, S_n) + AZ_{n+1}, \tag{23}$$

with m_{WS} *given in (20). Further, it holds*

$$\begin{aligned} W_{n+1} &= \exp\left(m_{n+1}^W(W_n) + \Sigma_W Z_{n+1}^W\right), \\ S_{n+1} &= m_{n+1}^S(W_n, S_n) + \Sigma_S\left(\rho Z_{n+1}^W + \sqrt{1-\rho^2} Z_{n+1}^S\right). \end{aligned} \tag{24}$$

3.2. State Constraints

The operation and technical design of the underlying P2H system restrict the state and control variables. The state-dependent control constraints derived below in Section 3.3 result from the following constraints on the state variables.

Due to the configuration of the P2H system, the TES temperature is bounded, i.e., $R_n \in [r_{\min}, r_{\max}]$ for all n. More precisely, ensuring constant SG inlet and outlet temperatures by fluid bypass regulation implies that the storage temperature R_n cannot be greater than $r_{\max} = T^{\mathrm{SG,in}}$ and cannot fall below $r_{\min} = T^{\mathrm{SG,out}}$.

In contrast, the exogenous state variables are generally based on our modeling approach in (14). Wind speeds are by nature non-negative and potentially unbounded, implying that $W_n \in (0, \infty)$. Unlike wind speeds, the electricity prices are also allowed to become negative and we have $S_n \in (-\infty, \infty)$. A negative price may occur in the case of overproduction of electricity, while at the same time there is a lower demand in the grid. In addition, in this case, producers are penalized for feeding in additional power, while consumers are rewarded for using electrical energy from the grid.

3.3. Control Constraints

The various state and operational constraints mentioned in the previous subsections imply constraints on the control and lead to state-dependent sets of feasible controls from which the controller can select the actions. In particular, the control A_n for the period $[t_n, t_{n+1})$ can only be selected such that technical upper limits for the HTHX outlet temperature $T_n^{\mathrm{HT,out}} = \tau_{\max}^{\mathrm{HT,out}}$ and the HTHX inlet temperature $T_n^{\mathrm{HT,in}} = \tau_{\max}^{\mathrm{HT,in}}$ are not exceeded. While the maximal HTHX outlet temperature $\tau_{\max}^{\mathrm{HT,out}}$ is directly related to the maximal compressor shaft speed, the maximal inlet temperature $\tau_{\max}^{\mathrm{HT,in}}$ is set by system constraints of the HTHP. Furthermore, the controller needs to consider that the time-varying charging and discharging factors $l^{\mathrm{C}}, l^{\mathrm{D}}$, which determine the bypasses, can only take values in $[0, 1]$ and must ensure that the TES temperature R, which is also time-varying, does not leave the range $[r_{\min}, r_{\max}] = [T^{\mathrm{SG,out}}, T^{\mathrm{SG,in}}]$. Given that the state at the beginning of the period $[t_n, t_{n+1})$ is $X_n = x = (r, w, s)$, the control A_n can be selected from a set of feasible controls

$$\mathcal{A}_n(x) = [\underline{a}(r), \overline{a}(r)] \subset \mathcal{A}, \tag{25}$$

where $\underline{a}(r), \overline{a}(r)$ are piecewise linear functions of the TES temperature, which we derive in Equations (S4.3) and (S.4.6). Figure 4 illustrates the derived set of feasible controls and their dependence on R for the system parameters listed in the Supplementary Material, Section S2. More precisely, we assume that $\dot{m} = 6$ kg/s is constant, which implies $T^{\mathrm{SG,out}} = 185.8$ °C and $T^{\mathrm{SG,in}} = 303$ °C and limits the TES operating temperature downwards and upwards, respectively.

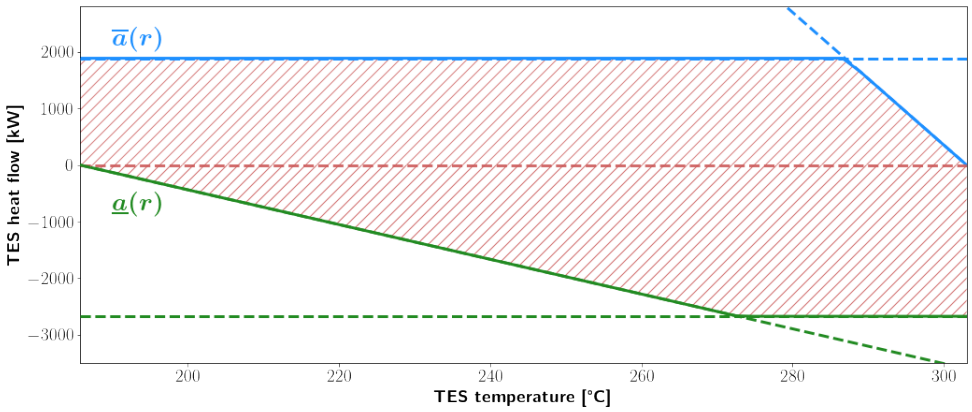


Figure 4. Visualization of the control constraints (25) as a function of the TES temperature. Upper bound $\overline{a}(r)$ (blue), lower bound $\underline{a}(r)$ (green), and the sets of feasible controls for the control $A_n(x)$ (red). If the TES is almost full, the upper bound $\overline{a}(r)$ is decreasing and approaches zero to prevent overheating during charging. If the TES is not sufficiently full, the decreasing lower limit $\underline{a}(r)$ prevents undercooling during discharging. In both cases, the heat flow is throttled accordingly. The maximum of the positive upper bound $\overline{a}(r)$ and the minimum of the negative lower bound $\underline{a}(r)$ result from the maximal inlet and outlet temperatures of the HTHX.

3.4. Operational Costs

The operational costs of the system are directly related to the HTHP's electricity consumption P^{H}, which is linked to the HTHX outlet and inlet temperature $T^{\mathrm{HT,out}}$ and $T^{\mathrm{HT,in}}$ chosen by the controller, see (11). Covering the electricity demand depends on the available power output P^{W} generated by the WT, which in turn is a function of the wind speed W. The difference $P^{\mathrm{G}} = P^{\mathrm{H}} - P^{\mathrm{W}}$ must be drawn from the grid at the price S if $P^{\mathrm{G}} > 0$. We assume that P^{W} is free of charge and does not incur any additional costs such as operational and maintenance costs. Consequently, only the consumed grid power P^{G} must be paid and incurs costs. A negative P^{G} means an overproduction of WT power that can be sold to the grid for revenue. Here, the selling price is usually lower than the purchase price S.

Running Cost.

In our model, we consider the running operational costs $C_n : \mathcal{X} \times \mathcal{A} \to \mathbb{R}$ in each of the periods $n = 0, \ldots, N-1$. These are defined as the expected cumulative costs in the period $[t_n, t_{n+1})$, given that at time t_n the state $X_n = x = (r, w, s)^\top$ and the control $A_n = a$ is chosen, and read as

$$C_n(x, a) = \mathbb{E}_{n,x,a}\left[\int_{t_n}^{t_{n+1}} \Psi(t, W(t), S(t), a)\mathrm{d}t\right], \tag{26}$$

where Ψ is defined by

$$\Psi(t, W(t), S(t), a) = S(t)(\pi^{\mathrm{H}}(a) - P^{\mathrm{W}}(t))^+ - \zeta S_{\mathrm{sell}}(t)(\pi^{\mathrm{H}}(a) - P^{\mathrm{W}}(t))^-, \tag{27}$$

with π^{H} introduced in (12). The conditional expectation $\mathbb{E}_{n,x,a}(\cdot) = \mathbb{E}(\cdot|X_n = x,\ A_n = a)$ emphasizes the dependence on the current time grid point t_n and the current state $X_n = x$ as well as the action $A_n = a$ selected at this state. Further, we denote by $z^+ = \max(z, 0)$ and $z^- = \max(-z, 0)$ the positive and negative part of $z \in \mathbb{R}$, respectively. The functional Ψ in (27) is divided into two parts as follows: (i) the costs for buying electricity from the grid at price $S(t)$ and (ii) the revenue for selling excess energy at a lower price $S_{\mathrm{sell}}(t)$. Here, $\zeta \in \{0, 1\}$ is a user defined model parameter that indicates if selling is allowed or not. If it is not allowed to sell excess energy to generate revenue, we set $\zeta = 0$. In this case, the surplus or overproduction of energy is discarded. For $\zeta = 1$, energy is fed into the grid for a reduced market price S_{sell} given by

$$S_{\mathrm{sell}}(t) = S(t) - \eta(t), \tag{28}$$

where $\eta : [0, t_{\mathrm{E}}] \to \mathbb{R}^+$ is called spread. This spread reflects transaction fees, taxes or the willingness of the grid operator to buy energy only at a certain discount on the market price $S(t)$. A special situation occurs in times of negative market prices S, which are often caused by energy overproduction. In this case, buying from the grid leads to a reward, i.e., one gets paid for purchasing energy. Selling, on the other hand, causes additional cost to keep the grid stable due to the abundance of energy. Here, a spread $\eta > 0$ leads to a further reduction of the selling price, which results in higher costs for feeding energy into the grid and therefore makes selling less attractive. For more details on the computation of the running costs $C_n(x, a)$ and particularly the conditional expectation in (26), see Section S7.1 of the Supplementary Material.

Terminal Cost.

At the end of the planning period, a terminal cost function $G_N : \mathcal{X} \to \mathbb{R}$ can be used to evaluate the terminal state of the system, particularly the amount of thermal energy stored in the TES, in monetary terms. A typical example are *penalty and liquidation payments* that are applied if the TES medium temperature is below or above a certain user-defined critical value $r_{\mathrm{crit}} \in [r_{\min}, r_{\max}]$. Suppose that the terminal state is $X_N = x = (r, w, s)^\top$, then the terminal cost is defined by

$$G_N(x) = \begin{cases} g_{\mathrm{Pen}}(r) s_{\mathrm{Pen}}, & r < r_{\mathrm{crit}}, \\ g_{\mathrm{Liq}}(r) s_{\mathrm{Liq}}, & r \geq r_{\mathrm{crit}}, \end{cases} \tag{29}$$

where the functions $g_{\mathrm{Pen}} : \mathcal{X} \to \mathbb{R}^+$ and $g_{\mathrm{Liq}} : \mathcal{X} \to \mathbb{R}^-$ describe the amount of thermal energy required to adjust the TES temperature from r to r_{crit}. In the case of penalization, g_{Pen} units of thermal energy must be fed in, while for liquidation, g_{Liq} units are withdrawn. If the TES is not sufficiently filled, i.e., $r < r_{\mathrm{crit}}$, a penalty is applied for energy consumption at a fixed price $s_{\mathrm{Pen}} \geq 0$, depending on the respective temperature difference. In the opposite case, excess energy in the TES is liquidated, which means that energy is sold at the fixed price $s_{\mathrm{Liq}} \geq 0$, which generates a revenue that appears as a negative terminal cost G_N. It should be noted that this definition of the terminal cost includes a worthless expiration of the TES by setting $s_{\mathrm{Pen}} = s_{\mathrm{Liq}} = 0$.

3.5. State and Action Space

Summarizing all the information from above, the state process $X_n \in \mathcal{X}$ of the P2H system at time t_n is described by $X = (X_n)_{n=0,\dots,N}$ with

$$X = (R, W, S), \tag{30}$$

where the state space $\mathcal{X} \subset \mathbb{R}^3$ is defined as

$$\mathcal{X} = [r_{\min}, r_{\max}] \times (0, \infty) \times (-\infty, \infty), \tag{31}$$

with the boundaries according to $r_{\min} = T^{\mathrm{SG,out}}$ and $r_{\max} = T^{\mathrm{SG,in}}$, resulting from system-related and technical constraints as well as model assumptions. The control process $A = (A_n)_{n=0,\dots,N-1}$ at a given state X_n is specified by $A_n = u_n(X_n) \in \mathcal{A}$ with decision rules

$$u_n : \mathcal{X} \to \mathcal{A}, \quad x \mapsto u_n(x) \in \mathcal{A}_n(x), \quad n = 0, \dots, N-1. \tag{32}$$

The sequence $u = (u_n)_{n=0,\dots,N-1}$ of decision rules is called a policy. Moreover, the system at a state $X_n = x$ can be controlled by choosing the action $u_n(x) = a$. The set of feasible controls $\mathcal{A}_n(x) \subset \mathcal{A}$ in state $x \in \mathcal{X}$ at time t_n is based on the derived control constraints (25) in Section 3.3 and reads for $x = (r, w, s)$ as

$$\mathcal{A}_n(x) = [\underline{a}(r), \overline{a}(r)]. \tag{33}$$

3.6. Transition Operator

The transition from one state to another, within the feasible set $\mathcal{X}$, is mathematically described by the transition operator. For state X_n, action $A_n = u_n(X_n)$ at time point t_n, and a random disturbance $Z_{n+1} = (Z^W_{n+1}, Z^S_{n+1}) \sim \mathcal{N}(0_2, \mathrm{I}_2)$, the state dynamics of the system is defined by the transition operator as

$$X_{n+1} = \mathcal{T}_n(X_n, A_n, Z_{n+1}). \tag{34}$$

In this context, according to (3), the endogenous state dynamics for the TES temperature is given by $R_{n+1} := g^R(X_n, A_n)$ with

$$g^R(x, a) = r + \frac{1}{m_{\mathrm{s}} c_{\mathrm{S}}} a \Delta t, \quad \text{for} \quad x = (r, w, s). \tag{35}$$

The wind speed W as an exogenous, stochastic state is modeled as an exponential discrete-time Ornstein–Uhlenbeck process, see (24). Its dynamic reads as

$$W_{n+1} := g^W_{n+1}(X_n, Z_{n+1}) \quad \text{with} \quad g^W_{n+1}(x, z) = \exp(m^W_{n+1}(w) + \Sigma_W z^W), \tag{36}$$

for $x = (r, w, s)$ and $z = (z^W, z^S)$. The electricity price S, the second exogenous and stochastic state, is described using (24) with

$$S_{n+1} := g^S_{n+1}(X_n, Z_{n+1}) \quad \text{with} \quad g^S_{n+1}(x, z) = m^S_{n+1}(w, s) + \Sigma_S(\rho z^W + \sqrt{1 - \rho^2} z^S). \tag{37}$$

Putting everything together, the transition operator (34) is given by

$$\mathcal{T}_n(x, a, z) = \Big(g^R(x, a),\ g^W_{n+1}(x, z),\ g^S_{n+1}(x, z) \Big). \tag{38}$$

For the discrete-time system we will consider a filtered probability space with the filtration $\mathbb{F} = (\mathcal{F}_n)_{n=0,\dots,N}$ where the σ-algebras $\mathcal{F}_n = \sigma(Z_1, \dots, Z_n)$ are generated by the independent random variables $Z_1, \dots, Z_n$, and $\mathcal{F}_0 = \{\emptyset, \Omega\}$ is the trivial σ-algebra.

3.7. Performance Criterion and Optimization Problem

The combination of the discrete-time system and the formulation of the corresponding cost functional allows us to determine the operational performance criterion. In this context, the optimal control of the system is related to a cost-optimal policy such that the expected

aggregated running costs (26) for operating the P2H system and the terminal costs (29) for an initial state $X_0 = x \in \mathcal{X}$ are minimized. A policy $u = (u_n)_{n=0,\dots,N-1}$ is a sequence of decision rules $u_n : \mathcal{X} \to \mathcal{A}$ that maps a given state $x \in \mathcal{X}$ to an admissible action $a \in \mathcal{A}_n(x)$. At each point of time, the associated objective function or performance criterion $J_n^u : \mathcal{X} \to \mathbb{R}$ is given by

$$J_n^u(x) = \mathbb{E}\left[\sum_{i=n}^{N-1} C_i(X_i, u_i(X_i)) + G_N(X_N) \,\middle|\, X_n = x \right], \tag{39}$$

with the running and terminal cost defined in Section 3.4. We denote by $\mathcal{U}$ the set of all admissible policies u such that the objective function (39) is well-defined and $A_n = u_n(X_n)$ satisfies the control constraints for all $n = 0, \dots, N-1$. An admissible policy $u \in \mathcal{U}$ is called optimal if

$$J_n^{u^*}(x) = V_n(x) = \inf_{u \in \mathcal{U}} J_n^u(x). \tag{40}$$

The function V_n is called the *value function* and describes the minimal expected aggregated running costs. Therefore, finding the value function is equivalent of finding the optimal policy u^*.

Recursion Property and Bellman Equation.

In practice, it is not tractable to minimize over the space of all admissible policies $\mathcal{U}$. We will see that the performance criterion (39) satisfies a recursion property from which we are able to deduce an alternative way for obtaining the value function. The following theorems can be found in Bäuerle and Rieder ([21], pp. 21–23). Theorem 1 states that the objective function (39) fulfills a recursion property.

Theorem 1 (Recursion property). *Let $u = (u_n)_{n=0,\dots,N-1}$ be a fixed policy. Then for every $n = 0, \dots, N-1$ and $x \in \mathcal{X}$ the objective function $J_n^u(x)$ satisfies*

$$\begin{aligned} J_N^u(x) &= G_N(x), \\ J_n^u(x) &= C_n(x, a) + \mathbb{E}_{n,x,a}[J_{n+1}^u(X_{n+1})]. \end{aligned} \tag{41}$$

Using Theorem 1, we obtain that the value function $V_n(x)$ is the solution of the well-known *Bellman equation*.

Theorem 2 (Bellman equation). *For every $x \in \mathcal{X}$, the value function $V_n(x)$ for $n = 0, \dots, N$ satisfies the Bellman equation*

$$\begin{aligned} V_N(x) &= G_N(x), \\ V_n(x) &= \inf_{a \in \mathcal{A}_n(x)} \Big\{ C_n(x, a) + \mathbb{E}_{n,x,a}[V_{n+1}(X_{n+1})] \Big\}. \end{aligned} \tag{42}$$

The Bellman equation reduces the problem of finding an optimal policy $u^* \in \mathcal{U}$ to a recursion in which only the optimal actions for each time point n must be found.

(Optimal) State Action Function and Properties.

Many applications and algorithms use an alternative performance criterion for J_n^u, which is often called the state action function and is denoted by Q_n^u [36]. It is defined for all $(x, a) \in \mathcal{X} \times \mathcal{A}_n(x)$ as

$$Q_n^u(x,a) = \mathbb{E}_{n,x,a}\left[\sum_{i=n}^{N-1} C_n(X_i, u_i(X_i)) + G_N(X_N)\right]. \tag{43}$$

Note, that the expectation is conditional to $X_n = x, u_n(x) = a$, and given a policy u with deterministic decision rules u_n, it holds that

$$J_n^u(x) = Q_n^u(x, u_n(x)), \tag{44}$$

for all $n = 0, \ldots, N$. As the name suggests, the function Q_n^u assigns a value to each feasible state action pair (x, a), instead of a single value for each state x as is the case with the performance criterion J_n^u. By Theorem 1, we get that Q_n^u, satisfies the recursion

$$\begin{aligned} Q_n^u(x,a) &= C_n(x,a) + \mathbb{E}_{n,x,a}[J_{n+1}^u(X_{n+1})] \\ &= C_n(x,a) + \mathbb{E}_{n,x,a}[Q_{n+1}^u(X_{n+1}, u_{n+1}(X_{n+1}))]. \end{aligned} \tag{45}$$

The optimal state action function $Q_n^*(x, a)$ for $(x, a) \in \mathcal{X} \times \mathcal{A}_n(x)$ is given by

$$Q_n^*(x,a) = \inf_{u \in \mathcal{U}} Q_n^u(x,a). \tag{46}$$

It further relates to the original value function $V_n(x)$ by

$$V_n(x) = \inf_{a \in \mathcal{A}_n(x)} Q_n^*(x,a), \tag{47}$$

with optimal policy $u^* = (u_n^*)_{n=0,\ldots,N-1}$ and solves the Q-version of the Bellman equation [37] below.

Theorem 3 (Q-version Bellman equation)**.** *For every $(x, a) \in \mathcal{X} \times \mathcal{A}_n(x)$, the optimal state action function $Q_n^*(x, a)$ for $n = 0, \ldots, N$ satisfies*

$$\begin{aligned} Q_N^*(x,a) &= G_N(x), \\ Q_n^*(x,a) &= C_n(x,a) + \mathbb{E}_{n,x,a}\left[\inf_{a' \in \mathcal{A}_{n+1}(X_{n+1})} Q_{n+1}^*(X_{n+1}, a')\right]. \end{aligned} \tag{48}$$

The main difference between the Bellman equation (42) and the Q-version (48) is that the expectation and minimization are interchanged. This offers certain computational advantages both in the calculation of the expected value and in the minimization, which is now a deterministic problem.

4. Backward Dynamic Programming

In this section, we introduce an algorithm that solves the stochastic optimal control problem. This method is inspired by the Bellman equation (42) and is known as backward dynamic programming (BDP). We will also discuss issues that naturally arise when using BDP and propose techniques on how to address them.

4.1. Backward Recursion Algorithm

Note that, given a state $X_n = x$ and action $A_n = a$, the next state X_{n+1} is obtained by the transition operator (34), which depends on the random disturbance Z_{n+1}. Furthermore, the randomness in the subsequent state is only induced by the disturbance allowing to take the expectation with respect to the random variable Z_{n+1}, instead of X_{n+1} leading to

$$\mathbb{E}_{n,x,a}[V_{n+1}(X_{n+1}))] = \mathbb{E}[V_{n+1}(\mathcal{T}_n(x, a, Z_{n+1}))]. \tag{49}$$

Algorithm 1 summarizes the backward recursion procedure of BDP, which will be used to solve the MDP derived in Section 3.

Algorithm 1 Backward dynamic programming

1: Initialize $V_N(x) = G_N(x)$ for all $x \in \mathcal{X}$ and set $n = N - 1$
2: Compute for all $x \in \mathcal{X}$ the value function at time n

$$V_n(x) = \inf_{a \in \mathcal{A}_n(x)} \Big\{ C_n(x, a) + \mathbb{E}[V_{n+1}(\mathcal{T}_n(x, a, Z_{n+1}))] \Big\}$$

and the associated optimal decision rule by

$$u_n^*(x) \in \arg\min_{a \in \mathcal{A}_n(x)} \Big\{ C_n(x, a) + \mathbb{E}[V_{n+1}(\mathcal{T}_n(x, a, Z_{n+1}))] \Big\}$$

3: If $n > 0$ set $n = n - 1$ and go to step 1 else stop the algorithm.

4.2. Approximate Solution of the Bellman Equation

BDP may face some issues. For instance, if the state and action space are large or high-dimensional, it may suffer from the curse of dimensionality, or in the case of continuous spaces, the optimization problem in step 2 of Algorithm 1 may be difficult to solve. Another practical issue is the calculation of the expected value for all $x \in \mathcal{X}$, which may be computationally intractable due to the unavailability of closed-form expressions. Since it may be difficult to solve the Bellman equation exactly, we need to make certain simplifications in order to solve the issues described above.

State and Action Space Discretization.

Firstly, the state space $\mathcal{X} \subset \mathbb{R}^3$ given in (31) is discretized into distinct grid points. The value function is then calculated and saved for the given reference grid points. For the discretization, the seasonalities μ_W and μ_S of the Ornstein–Uhlenbeck processes (14) are used to construct time-varying sets of grid points. The advantage of introducing this time dependence is that the value function is only calculated for regions of interests, i.e., subsets of $\mathcal{X}$ that are more likely to appear at certain times. This leads to the family of discretized state spaces

$$\widetilde{\mathcal{X}}_n = \{r_1, \ldots, r_{n_R}\} \times \{w_{1,n}, \ldots, w_{n_W,n}\} \times \{s_{1,n}, \ldots, s_{n_S,n}\}, \quad n = 0, \ldots, N-1, \tag{50}$$

with $n_R, n_W, n_S \in \mathbb{N}$. The specific choice of grid points used in our calculations can be found in Section S8 of the Supplementary Material. Note, that the discretization points for the TES temperature is the same at every time point and only those of wind speed and electricity price change. The discrete structure of the state space $\widetilde{\mathcal{X}}_n$ allows us to solve the Bellman equation for each $x \in \widetilde{\mathcal{X}}_n$ separately. To solve the problem of minimization over the action space $\mathcal{A}_n(x) \subset \mathbb{R}^2$ in BDP, the action space $\mathcal{A}_n(x)$ is also discretized into grid

points $\widetilde{\mathcal{A}}_n(x) = \{a_1, \ldots, a_{n_A}\}$ for a given $n_A \in \mathbb{N}$. Hence, the minimization consists of calculating a value for each action and picking the action with the smallest value.

Approximation of the Expected Value.

Given a state $X_n = x$ and action $A_n = a$, the conditional expected value in the Bellman equation with respect to the next state X_{n+1} is given by an unconditional expectation with respect to the random disturbance Z_{n+1}, see (49). Let $\mathcal{Z} = \{z_1, \ldots, z_L\}$ be a set of value that Z_{n+1} can take and denote $\widehat{Z}_{n+1}$ as the discrete random variable taking values in $\mathcal{Z}$. Further, let $p_l = \mathbb{P}(\widehat{Z}_{n+1} = z_l)$ be the corresponding probability that $\widehat{Z}_{n+1}$ takes value z_l; then the expected value can be approximated as weighted sum

$$\mathbb{E}[V_{n+1}(\mathcal{T}_n(x, a, Z_{n+1}))] \approx \mathbb{E}[V_{n+1}(\mathcal{T}_n(x, a, \widehat{Z}_{n+1}))] = \sum_{l=1}^{L} p_l V_{n+1}(\mathcal{T}_n(x, a, z_l)). \tag{51}$$

The set $\mathcal{Z}$ is called a *quantizer* of Z_{n+1} and defines a partition on $\mathbb{R}^2$ into L subsets $\mathcal{C}(z_l)$, $l = 1, \ldots, L$, where each point z_l is uniquely assigned to a subset. As a consequence the probability p_l corresponds to the probability that Z_{n+1} takes values in $\mathcal{C}(z_l)$, more precisely

$$p_l = \mathbb{P}(Z_{n+1} \in \mathcal{C}(z_l)). \tag{52}$$

The calculation of these probabilities often requires solving high-dimensional integrals over the subsets $\mathcal{C}(z_l)$ with respect to the density of Z_{n+1}. Below, we will explain how to numerically obtain these probabilities. But before we come to this, we want to point out a practical problem that is caused by the quantizer $\mathcal{Z}$.

Interpolation and Extrapolation.

In general, the states $x_l = \mathcal{T}_n(x, a, z_l)$ do not coincide with the grid points in $\widetilde{\mathcal{X}}_{n+1}$ for which the value function V_{n+1} is calculated and saved. If the points are allocated in between existing values, an interpolation can be used to determine $V_{n+1}(x_l)$; otherwise, this value must be calculated by extrapolation. In this paper, we use linear interpolation if $x_l = \mathcal{T}_n(x, a, z_l)$ is in between existing point and extrapolate with the value of the nearest neighbor in the set $\widetilde{\mathcal{X}}_{n+1}$. The corresponding extrapolation errors are usually larger than those resulting from interpolation. In any case, the value $V_{n+1}(x_l)$ is weighted by the probability p_l, and if this is small, the corresponding error introduced will be scaled down by p_l. For this reason, the probabilities p_l can be used to reduce and control the errors in the calculation of the value function. We also note that an appropriate choice of the discretized state space $\widetilde{\mathcal{X}}_{n+1}$ helps to mitigate extrapolation. Details on the construction of the discretization that takes this fact into account can be found in Section S8 of the Supplementary Material.

Remark 1. *Due to the approximation of the expected value and action space discretization, the value function obtained is an approximation of V and will be denoted by $\widetilde{V}$. Therefore, the control corresponding to $\widetilde{V}$ is an approximation of the optimal control. The calculation of the value function $\widetilde{V}_n$ for $x \in \widetilde{\mathcal{X}}_n$ in the BDP Algorithm 1 reduces to*

$$\widetilde{V}_n(x) = \min_{a \in \widetilde{\mathcal{A}}_n(x)} \left\{ C_n(x, a) + \mathbb{E}[\widetilde{V}_{n+1}(\mathcal{T}_n(x, a, \widehat{Z}_{n+1}))] \right\}. \tag{53}$$

4.3. Optimal Quantizer for the Expected Value

The choice of the quantizer $\mathcal{Z}$ is important in order to obtain a good approximation of the expected value in (49). An approach to obtain them was proposed by Pagès [38,39], in which a so-called optimal quantizer $\mathcal{Z}^* = \{z_1^*, \ldots, z_L^*\}$ is selected. In the following, we will briefly discuss the optimality of this quantizer as well as some theoretical definitions and results. For more information, we refer the reader to the work of Pagès [38,39].

Optimal Quantizer of Z.

For a square integrable random variable Z in $\mathbb{R}^2$ with probability density of Z by f_Z, we denote the set $\mathcal{Z} = \{z_1, \ldots, z_L\} \subset \mathbb{R}^2$ and Borel-measurable function $q : \mathbb{R}^2 \to \mathcal{Z}$. The random vector $q(Z)$ is called an L-quantization of Z and $\mathcal{Z}$ is called an L-quantizer. The aim is to find an L-quantization q such that the quadratic distortion D_L^Z given by

$$D_L^Z(\mathcal{Z}) = \mathbb{E}(\| Z - q(Z) \|_2^2) \tag{54}$$

is minimized. It can be shown that the so-called Voronoi L-quantization defined by

$$q_{\text{Vor}}(z) = \sum_{l=1}^{L} z_l \mathbb{1}_{\mathcal{C}(z_l)}(z) \tag{55}$$

is minimizing D_L^Z, where $\mathcal{C}(z_l), l = 1, \ldots, L$ are Voronoi cells with

$$\mathcal{C}(z_l) \subset \{y \in \mathbb{R}^d | \ \| z_l - y \|_2 \leq \| z_i - y \|_2, \ i = 1, \ldots, L\}. \tag{56}$$

Note, that the Voronoi cells form a partition of $\mathbb{R}^2$. These quantizations can be understood as the nearest neighbor projection of Z onto the set $\mathcal{Z}$. We denote the Voronoi L-quantization of Z by $\widehat{Z} = q_{\text{Vor}}(Z)$. Moreover, the probability that $\widehat{Z}$ takes the value z_l is given by

$$p_l = \mathbb{P}(\widehat{Z} = z_l) = \mathbb{P}(Z \in \mathcal{C}(z_l)) = \int_{\mathcal{C}(z_l)} f_Z(z) \, \mathrm{d}z. \tag{57}$$

For the Voronoi L-quantization $\widehat{Z}$, the quadratic distortion can be written as

$$D_L^Z(\mathcal{Z}) = \mathbb{E}(\| Z - \widehat{Z} \|_2^2) = \sum_{l=1}^{L} \mathbb{E}(\mathbb{1}_{\mathcal{C}(z_l)}(Z) \| Z - z_l \|_2^2) = \int_{\mathbb{R}^2} f_Z(z) \min_{1 \leq l \leq L} \| z - z_l \|_2^2 \ \mathrm{d}z. \tag{58}$$

Now for $\widehat{Z}$ the mapping $\mathcal{Z} \mapsto D_L^X(\mathcal{Z})$ is continuous and yields a minimum $\mathcal{Z}^* = \{z_1^*, \ldots, z_L^*\}$ with distinct components [38]. This set is called an optimal L-quantizer and satisfies

$$D_L^Z(\mathcal{Z}^*) = \min_{\mathcal{Z} \subset \mathbb{R}^2} D_L^Z(\mathcal{Z}). \tag{59}$$

The existence of an optimal quadratic L-quantizer and the convergence are proven in [40]. In addition, Zardor's theorem [41] provides a prescribed level of accuracy for the number of quantization points L. Apart from an upper bound on the quadratic distortion with an error rate of order $L^{-1/d}$, where d is the dimension of the random variable (in our case $d = 2$), this theorem also establishes asymptotic convergence to zero as the number of quantization points L goes to infinity.

Calculation of Quantizers and Probabilities.

Numerical methods such as the Competitive Learning Vector Quantization (CLVQ) or (randomized) Llyods algorithm are often used to compute optimal quantizers, see [38,42,43]. For standard normally distributed random variables in $\mathbb{R}^d$, pre-calculated optimal quadratic L-quantizers for different $L, d \in \mathbb{N}$, with their corresponding probability mass of the Voronoi cells are available on www.quantize.maths-fi.com (accessed on 7 February 2026). Due to the accessibility of the high precision precalculated optimal quantizers $\mathcal{Z} = \{z_1^*, \ldots, z_L^*\}$ and probabilities p_l, $l = 1, \ldots, L$ of Z_{n+1}, we will use them in this paper. The optimal quadratic 200-quantizer $\mathcal{Z}^*$ of Z with its respective Voronoi cells and corresponding probability mass is shown in Figure 5.

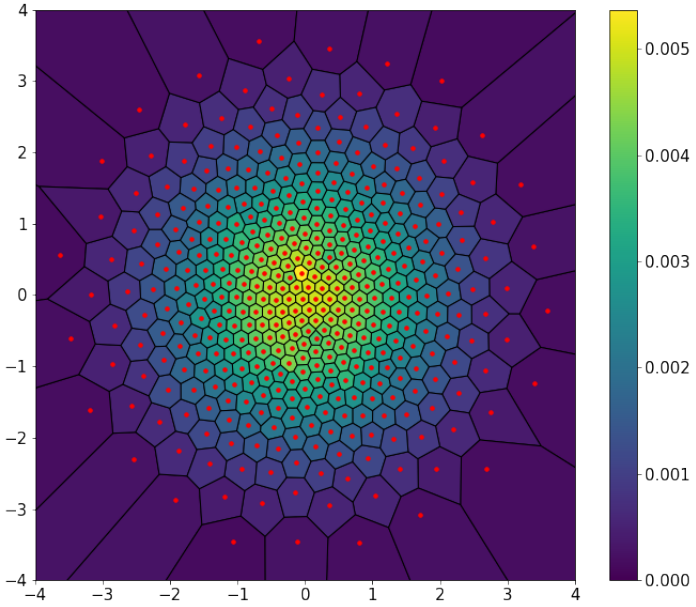


Figure 5. An optimal quadratic 200-quantizer (red dots) with Voronoi cells for a standard bivariate Gaussian random variable, taken from www.quantize.maths-fi.com (accessed on 7 February 2026). The color of the Voronoi cells indicates their corresponding probability mass.

Application to the Bellman Equation.

When applying the optimal quantization to (42), it is obvious we need to replace the continuous random variable Z by its optimal quantization $\widehat{Z}$, associated with the optimal quantizer $\mathcal{Z}^*$ in order to obtain a reasonable approximation of the expected value. However, $g(\widehat{Z})$ is in general not an optimal quantizer for $g(Z)$, when applying the nonlinear transformation $g(z) = V_{n+1}(\mathcal{T}_n(x, a, z))$. If g is bounded and continuous, then the convergence of $\mathbb{E}[\widehat{Z}]$ to $\mathbb{E}[Z]$ as the number of quantization points L grows to infinity, see [40], which implies the convergence $\mathbb{E}[g(\widehat{Z})] \to \mathbb{E}[g(Z)]$. If we make further assumptions on g, the convergence in the sense

$$\lim_{L\to\infty} L^{\alpha_g} |\mathbb{E}[g(Z)] - \mathbb{E}[g(\widehat{Z})]| \leq C_{g,Z}, \tag{60}$$

can be proven for different classes of functions g and precise convergence rates $\alpha_g > 0$ can be formulated; for more details see [44]. The constant $C_{g,Z}$ depends on the properties of g and the disturbance Z. In particular, if g is a Lipschitz continuous function with Lipschitz coefficient L_g, we obtain that

$$|\mathbb{E}[g(Z)] - \mathbb{E}[g(\widehat{Z})]| \leq L_g \mathbb{E}[\| Z - \widehat{Z} \|_1] \leq L_g \mathbb{E}[\| Z - \widehat{Z} \|_2] = L_g \sqrt{D_L^Z(\mathcal{Z}^*)}. \tag{61}$$

5. Reinforcement Learning Techniques

This section presents reinforcement learning algorithms that can tackle some of the problems mentioned in the context of BDP. We first introduce a quite general class of algorithms called temporal difference (TD) learning methods and then study Q-learning as a special case. Their aim is to approximate the value function in an appropriate parameter space and to construct the optimal policy with respect to this approximation. These methods rely on gradient descent to update the parameters with information obtained by generating samples of the controlled state process. In practice, the information for these methods does not have to come from an explicit model, as in our case. Instead, it can also be provided as data from an observed real world process, which is why these algorithms are often referred to as model-free.

5.1. Temporal Difference Learning

In the following, we use a function approximation to approximate $V_n(x)$ for all $x \in \mathcal{X}$ and $n = 0, \dots, N-1$. Let $\theta_n \in \mathbb{R}^p$ be a parameter vector that describes an approximation $\overline{V}_n$ of the exact value function V_n in terms of $p \in \mathbb{N}$ parameters $\theta_n^1, \cdots, \theta_n^p$

$$\overline{V}_n : \mathcal{X} \times \mathbb{R}^p \to \mathbb{R}, \tag{62}$$

such that $V_n(x) \approx \overline{V}_n(x, \theta_n)$. Let us give some typical examples of $\overline{V}_n(x, \theta_n)$ below.

Linear Function Approximation.

In this class of functions [27,45], $\overline{V}_n(x, \theta_n)$ is represented by a linear combination of basis or ansatz functions ϕ_i, $i = 1, \dots, p$ with $\phi_i : \mathcal{X} \to \mathbb{R}$ as

$$\overline{V}_n(x, \theta_n) = \sum_{i=1}^{p} \theta_n^i \phi_i(x). \tag{63}$$

The parameter vector $\theta_n \in \mathbb{R}^p$ corresponds to the coefficients of the linear combination. Polynomial ansatz functions, Fourier basis functions, or radial basis functions (RBF) are examples of function classes that are used for the linear function approximation $\overline{V}_n$.

Feedforward Neural Networks.

Feedforward neural networks (FNNs) are simple artificial neural networks and are a popular choice for nonlinear function approximators [46,47]. Essentially, they consist of affine-linear maps and nonlinear activation functions. Let $d_0 = \mathbb{R}^d$ and $d_L \in \mathbb{R}$ denote the input and output dimension of the FNN, then $\overline{V}_n(x, \theta_n)$ is represented by the recursion

$$\overline{V}_n(x, \theta_n) = A_L \rho(A_{L-1} \rho(\cdots \rho(A_1 x + b_1) \cdots) + b_{L-1}) + b_L), \tag{64}$$

where L is the number of layers, $A_l \in \mathbb{R}^{d_l \times d_{l-1}}$ and $b_l \in \mathbb{R}^{d_l}$, $l = 1, \dots, L$, are weights and biases for each layer with width $d_l \in \mathbb{N}$, and $\rho : \mathbb{R} \to \mathbb{R}$ is a nonlinear activation function that is applied component-wise. The parameter vector θ_n is the collection of all matrices A_l and vectors b_l. Examples for activation functions are the *sigmoid function* $\rho(x) = \frac{1}{1+e^{-x}}$ or the *rectified linear unit* (ReLU) $\rho(x) = \max\{x, 0\}$. Nowadays, FNNs are frequently used because it is known that they fulfill the universal approximation property [48], i.e., any continuous function can be approximated arbitrarily well.

TD-Learning Loss Functional.

The corresponding parameter update for TD-learning can be derived by minimizing a loss functional given by the expected squared distance

$$\mathcal{L}(\theta_n) = \frac{1}{2}\overline{\mathbb{E}}[(V_n(X_n) - \overline{V}_n(X_n, \theta_n))^2], \quad n = 0, \dots, N-1. \tag{65}$$

The underlying distribution in the expectation of loss (65) is called the state distribution and is used to sample states X_n. Normally, this distribution is chosen such that it reflects the importance of certain states that X_n can take, i.e., states that are of interest for the controller or are likely to appear. A natural choice would be the distribution of X_n or the so-called steady-state distribution [27]. For a given policy u, it describes the likelihood of X_n taking a specific state for a given initial state. Sampling from the steady-state distribution is realized by creating trajectories starting from an initial state, while following the policy u. Here, the initial state is sampled from a predetermined distribution, for example a uniform distribution over the state space $\mathcal{X}$. Minimizing the loss with respect to θ_n can be achieved by gradient descent, which leads to an iterative update rule

$$\theta_n^{k+1} = \theta_n^k - \alpha_n^k \nabla_{\theta_n} \mathcal{L}(\theta_n^k). \tag{66}$$

Here, k denotes the current iteration of the parameters and $\alpha_n^k > 0$ is the step size or learning rate. Note, that by interchanging the gradient with the expectation, we formally obtain

$$\nabla_{\theta_n} \mathcal{L}(\theta_n) = \overline{\mathbb{E}}[(V_n(X_n) - \overline{V}_n(X_n, \theta_n)) \nabla_{\theta_n} \overline{V}_n(X_n, \theta_n)]. \tag{67}$$

Thus, we can obtain samples for the gradient of the loss $\mathcal{L}(\theta_n)$ by using samples of X_n according to the state distribution in (65). To achieve a good and unbiased estimator for the expected value of the gradient, multiple realizations with batch size $M \in \mathbb{N}$ are used and averaged. The update rule (66) is therefore replaced by

$$\theta_n^{k+1} = \theta_n^k - \alpha_n^k \frac{1}{M} \sum_{j=1}^{M} \delta_n^j \nabla_{\theta_n} \overline{V}_n(x_n^j, \theta_n^k), \tag{68}$$

with $\delta_n^j = V_n(x_n^j) - \overline{V}_n(x_n^j, \theta_n^k)$. The iterative gradient update rule (68) is a special case of the Robbins–Monro algorithm [49] and is referred to as stochastic gradient descent (SGD) [50]. By applying (42), we get

$$\delta_n^j = \inf_{a \in \mathcal{A}_n(x_n^j)} \{C_n(x_n^j, a) + \mathbb{E}_{n, x_n^j, a}[V_{n+1}(X_{n+1})]\} - \overline{V}_n(x_n^j, \theta_n^k). \tag{69}$$

There are some problems that need to be addressed before a parameter update can be performed.

Practical and Computational Issues.

Firstly, since V_{n+1} is unknown, we need to replace it with an approximation. One way to do this is to use a Monte Carlo estimation of the performance criterion. Here, multiple trajectories starting from $X_n = x_n^j$ and thus multiple realizations of the performance criterion are obtained and averaged. However, the optimal policy is also unknown as it requires knowledge about the value function. Thus, the best choice for the policy is the (sub)optimal policy induced by the value function approximation.

A more common approach is *bootstrapping*, which avoids following trajectories with a possibly suboptimal policy. In doing so, the value function V_{n+1} is replaced by its

corresponding parameterization $\overline{V}_{n+1}$. This brings some computational advantages, but at the expense that convergence towards the value function can not always be shown, which will be discussed below.

Another issue arises from the replacement of the value function V_{n+1} and the minimization in (69) which is performed with respect to this approximation. If u is the policy induced by these value function approximations, the optimal control obtained by the associated decision rule is given by $a_n^j = u_n(x_n^j)$. Note, however, that this control may not be optimal in the sense that the minimum over $\mathcal{A}_n(x_n^j)$ is attained and therefore degrades the approximation.

Last but not least, as for BDP, there are several ways to calculate the expectation in (69). For BDP, a quantization approach is used, which could as well be used here. Nevertheless, a more common approach is to use a Monte Carlo simulation and use samples of X_{n+1}.

TD-Learning Update.

In practice, the TD-learning methods use bootstrapping to replace the V_{n+1} and one-sample Monte Carlo estimates instead of extensive calculations of the expectation. This is mostly motivated by the fact that calculations as well as simulation of the state process and evaluation of the optimal policy are time-consuming and therefore computationally intensive. Bootstrapping also offers the advantage to update the parameters θ_n immediately after observing samples of X_n, where Monte Carlo estimates of the performance criterion must wait until the trajectories end. Parameter updating is performed using samples

$$(x_n^j, a_n^j, x_{n+1}^j)_{j=1,\dots,M}, \quad n = 0, \dots, N-1, \tag{70}$$

with $a_n^j = u_n(x_n^j)$ and $x_{n+1}^j = \mathcal{T}_n(x_n^j, a_n^j, z_{n+1}^j)$, where z_{n+1}^j is a realization of Z_{n+1}. This results in the following TD-learning update for the parameters at time n

$$\theta_n^{k+1} = \theta_n^k - \alpha_n^k \frac{1}{M} \sum_{j=1}^{M} \delta_n^j \nabla_{\theta_n} \overline{V}_n(x_n^j, \theta_n^k), \tag{71}$$

with temporal difference

$$\delta_n^j = C_n(x_n^j, a_n^j) + \overline{V}_{n+1}(x_{n+1}^j, \theta_{n+1}^k) - \overline{V}_n(x_n^j, \theta_n^k). \tag{72}$$

The scalar δ_n can be interpreted as the change in information when moving from state x_n to x_{n+1}.

5.2. Q-Learning

Q-learning was first proposed by Watkins in 1989 [10] and is essentially a special class of TD-learning. The starting point here is the state action function $Q_n^u(x, a)$ given in (43) instead of the objective function $J_n^u(x)$ from (39). Analogous to the derivation of the TD-learning method, a parameter vector $\theta_n \in \mathbb{R}^p$ is used for all $n = 0, \dots, N-1$ to describe the parameterization

$$\overline{Q}_n(x, a, \theta_n) : \mathcal{X} \times \mathcal{A} \times \mathbb{R}^p \to \mathbb{R}, \tag{73}$$

in order to approximate $Q_n^*(x, a)$. Due to the relation in (47), an approximate value function can also be derived by

$$\overline{V}_n(x, \theta_n) = \min_{a \in \mathcal{A}_n(x)} \overline{Q}_n(x, a, \theta_n). \tag{74}$$

Again, the aim is to minimize the following loss functional

$$\mathcal{L}^{Q}(\theta_n) = \frac{1}{2}\overline{\mathbb{E}}[(Q_n^*(X_n, A_n) - \overline{Q}_n(X_n, A_n, \theta_n))^2]. \tag{75}$$

As for the TD-learning loss function (65), a sample state distribution is used. Note that $u_n(X_n) = A_n$ is a random variable, which emphasizes the choice of an action distribution to obtain samples for A_n. For a given state x, this distribution can be directly defined by a probability measure on $\mathcal{A}_n(x)$ or by a selection policy $u^S = (u_n^S)_{n=0,\dots,N-1}$, with $A_n = u_n^S(X_n)$. The relation (74) motivates us to choose u^S such that the optimal action is sampled frequently. A natural choice is a *greedy* selecting policy

$$u_n^S(x) = \min_{a \in \mathcal{A}_n(x)} \overline{Q}_n(x, a, \theta_n), \tag{76}$$

with respect to the current approximation. Analogous to TD-learning using (48) in loss (75) in combination with SGD, this results in an iterative Q-learning update for the parameters θ_n

$$\theta_n^{k+1} = \theta_n^k - \alpha_n^k \frac{1}{M} \sum_{j=1}^{M} \delta_n^j \, \nabla_{\theta_n} \overline{Q}_n(x_n^j, a_n^j, \theta_n^k), \tag{77}$$

with iteration counter k, step size $\alpha_n^k > 0$, and temporal difference

$$\delta_n^j = C_n(x_n^j, a_n^j) + \mathbb{E}_{n, x_n^j, a_n^j}\left[\inf_{a' \in \mathcal{A}_{n+1}(X_{n+1})} Q_{n+1}^*(X_{n+1}, a')\right] - \overline{Q}_n(x_n^j, a_n^j, \theta_n^k). \tag{78}$$

Q-Learning Update.

As above, we need to tackle similar problems to define a parameter update, such as replacing the unknown Q_{n+1}^* and evaluating the expected value in (78). The most common variant of Q-learning uses bootstrapping to replace Q_{n+1}^* and one-sample estimates for the expected value, which results in the temporal difference

$$\delta_n^j = C_n(x_n^j, a_n^j) + \inf_{a' \in \mathcal{A}_{n+1}(x_{n+1}^j)} \overline{Q}_{n+1}(x_{n+1}^j, a', \theta_{n+1}^k) - \overline{Q}_n(x_n^j, a_n^j, \theta_n^k). \tag{79}$$

Remark 2. *To ensure convergence for iterative stochastic approximation methods [36], like SGD, the step sizes α_n^k for all $n = 0, \dots, N-1$ need to satisfy the Robbins–Monro conditions [49]*

$$\sum_{k=0}^{\infty} \alpha_n^k = \infty \quad \textit{and} \quad \sum_{k=0}^{\infty} (\alpha_n^k)^2 < \infty. \tag{80}$$

However, as a consequence of bootstrapping in TD-learning and Q-learning, the resulting gradient estimates may differ from those of the original underlying loss function (65). These kind of methods are different from SGD and are called semi-gradient methods and require separate convergence results as well as additional assumptions. It should be noted that the main convergence analysis for TD-learning and Q-learning is based on MDPs with infinite time horizon. Convergence results for linear function approximation (63) are provided in [51,52]. Nonlinear function approximators such as neural networks require additional assumptions and techniques like projection [53] or linearization [54] to guarantee convergence. Although these convergence results hold for infinite-horizon MDPs, they can still be applied to the finite-horizon case by augmenting the state with time as an additional state variable. Provided that we can formulate an equivalent problem with the augmented state, the aforementioned convergence results can be applied.

Without augmentation, convergence results for finite-horizon TD-learning and Q-learning are derived in [55] using linear and nonlinear function approximations. These results are based on the

recursive properties of the value functions and therefore require less restrictive assumptions than the infinite-horizon setting.

ε-Greedy Selection Policies.

Given a state $X_n = x$, the selection policy u^S is used to create samples of actions that ensure reasonable exploration of the action space $\mathcal{A}_n(x)$ and the state space of X_{n+1}. In the following, we discuss a commonly used class of selection policies that differ from the greedy policy (76). The greedy policy has one major disadvantage that can lead to poor approximations. If this greedy strategy is fully exploited, all decisions are based on the current function approximation, which itself is biased by approximation errors. This can lead to suboptimal action samples in the sense that the optimal action may not be sampled frequently. As a result, the approximation of $\overline{Q}$ will be poor as well as the value function approximation associated with it. We can compensate this by sampling actions from a uniform distribution on $\mathcal{A}_n(x)$. This approach has the advantage that all actions are selected with equal probability and an overall better approximation can be obtained for all actions. A drawback, however, is that the optimal action may again not be sampled very often. In the literature, this problem of choosing an appropriate u^S is known as the exploration-exploitation dilemma. Here, we want to exploit the optimal policy induced by the function approximation as much as possible while still sampling a reasonable number of other actions to not miss out on more optimal actions. In practice, a combination of random and greedy policies is used, known as *ε-greedy policy* with exploration rate $\varepsilon \in [0,1]$. Here, an action is either drawn from an uniform distribution on $\mathcal{A}_n(x_n)$ with probability ε or with probability $1-\varepsilon$ selected greedily as in (76). The exploration rate could be given by a simple linearly decaying scheme with $\varepsilon^k = \frac{\varepsilon_0}{k}$ for $\varepsilon_0 \in [0,1]$, where k denotes the iteration counter.

Remark 3. *In Powell [24], examples of exploration rates and step sizes are discussed. It is also mentioned that choosing an appropriate step size that satisfies the Robbins–Monro conditions (80) is hard in practice. It may happen that the step sizes decrease too quickly, so the parameters converge to a non-optimal solution. Hence, it is suggested to use small constant step sizes $\alpha_n^k = \alpha_0 > 0$ as it has been empirically observed that these work well in applications, although the second condition in (80) is violated.*

5.3. Experience Replay

Q-learning faces the problem that creating trajectories and samples can be a time-consuming task, especially if the dynamics of the system are difficult to simulate or the time horizon is large. This makes it intractable to use generated samples only once and then throw them away when the parameter update is completed. We will use a technique called *experience replay* [56,57] that solves the problem of wasting generated samples. This uses a so-called replay memory or replay buffer $\mathcal{R}$ with size $\mathcal{N}_R \in \mathbb{N}$ to store samples $(x_n, a_n, x_{n+1}) \in \mathcal{R}$ for $n = 0, \ldots, N-1$ and replay (reuse) them as needed in batches $(x_n^j, a_n^j, x_{n+1}^j)$, $j = 1, \ldots, M$ to update the parameters. Sampling past experience, for example, from a uniform distribution on $\mathcal{R}$, also helps to overcome the exploration-exploitation dilemma. Here, the samples obtained in the early phase of the algorithm are used repeatedly, making the choice of the exploration rate less important for the performance of the algorithm. We summarize the Q-learning method with replay buffer in Algorithm 2.

Calculating the minimum over all actions might be hard to accomplish if the action space is continuous, as in our case. In our numerical experiments, the action space $\mathcal{A}_n(x)$ is discretized as for BDP, which was explained in Section 4. The titles of the boxes are hyperlinks and can be used to navigate to the corresponding section in the article.

Algorithm 2 Q-Learning with Replay Buffer

1: Initialize $(\theta_n^0)_{n=0}^{N-1}$; set the maximum number of iterations $k^{\max} > 0$ and $k = 0$, batch size M; choose a selection policy $u^S = (u_n^S)_{n=0}^{N-1}$
2: Set $n = 0$; choose the initial states $x_0 \in \mathcal{X}$
while $n < N$ **do**
Select an action according to $a_n = u_n^S(x_n)$, observe x_{n+1} and store (x_n, a_n, x_{n+1}) in $\mathcal{R}$.
Sample batch $(x_n^j, a_n^j, x_{n+1}^j)_{j=1}^M$ from the replay buffer $\mathcal{R}$
for $j = 1, \ldots, M$ **do** calculate δ_n^j
if $n < N - 1$ **then**

$$\delta_n^j = C_n(x_n^j, a_n^j) + \min_{a \in \mathcal{A}_{n+1}(x_{n+1}^j)} \overline{Q}_{n+1}(x_{n+1}^j, a, \theta_{n+1}) - \overline{Q}_n(x_n^j, a_n^j, \theta_n^k)$$

else set $\delta_n^j = C_{N-1}(x_{N-1}^j, a_{N-1}^j) + G_N(x_N^j) - \overline{Q}_{N-1}(x_{N-1}^j, a_{N-1}^j, \theta_{N-1}^k)$
end if
end for
Choose $\alpha_n^k \in [0, 1]$ and update parameters

$$\theta_n^{k+1} = \theta_n^k - \alpha_n^k \frac{1}{M} \sum_{j=1}^{M} \delta_n^j \nabla_{\theta_n} \overline{Q}_n(x_n^j, a_n^j, \theta_n^k)$$

set $n = n + 1$
end while
3: Set $k = k + 1$; if $k = k^{\max}$ go to step 4; else go to step 2
4: Obtain optimal control for $x \in \mathcal{X}$ and $n = 0, \ldots, N-1$: $u_n^*(x) \in \arg\min_{a \in \mathcal{A}_n(x)} \overline{Q}_n(x, a, \theta_n^{k^{\max}})$

6. Numerical Results

In this section, the numerical results obtained by Algorithms 1 and 2, i.e., BDP and Q-learning, are presented and compared with each other. More precisely, we compare the results in terms of accuracy and computational effort of the computed solutions (value function and trajectory). A time horizon of $t_{\mathrm{E}} = 120$ h is selected for the numerical simulation to find the value function and optimal operation for the industrial P2H system during a working week (5 days). The system and algorithm parameters can be found in Tables S1a and S1b, respectively. To keep the results of the proposed algorithms comparable, the same action space discretization is used for both. The evaluation of the minimization over the discretized action space is done by calculating all action values and selecting the action with the minimal value. Penalty costs are applied at terminal time if the TES temperature is below a certain threshold value. In the experiments, the storage must be at least half-full, i.e., the critical value is set to $r_{\mathrm{crit}} = (r_{\max} - r_{\min})/2$. Furthermore, falling below r_{crit} is penalized with a penalization price $s_{\mathrm{Pen}} = 90$ €/MWh. We do not reward the liquidation of the TES energy and therefore set $s_{\mathrm{Liq}} = 0$ €/MWh. Selling excess energy into the grid is also not allowed and we set $\zeta = 0$. For the calculation of the value function, both methods made use of a time-varying state space discretization, see Section S8 of the Supplementary Material. For visualization and convenience, we display the value function for fixed grid points selected from the set $[r_{\min}, r_{\max}] \times [2, 23.5] \times [15, 55]$. This set is chosen such that it is a subset of $\mathcal{X}_n$ for all $n = 1, \ldots, 120$. In order to visualize the results with respect to the three-dimensional state space, we will fix some state variables to specific values and plot them against the remaining variables. In the following, these fixed values are chosen as the corresponding centers of the r-, w- and s-axes.

6.1. Backward Dynamic Programming

Let us first discuss the results of the value function computed by BDP, which are presented in Figure 6a. The graphics on the left and right show the value function for the initial time step $n = 0$ and the terminal costs $n = 120$, respectively, as a function of TES temperature and wind speed. Obviously, low wind speed and low TES temperature lead to higher expected costs. While the dependence on the storage temperature is almost linear, the wind speed has a significant nonlinear influence on the value function, whereby the latter effect are induced by the WT power curve model. In addition, the dependence of the value function on storage temperature and electricity price as well as on wind speed and electricity price for the initial time is depicted in Figure 6b on the left and right, respectively. The electricity price affects the expected costs mostly linearly with respect to the TES temperature, with higher prices leading to higher values. Again, higher storage temperatures reduce costs and compensate for expensive grid electricity. The relation between wind speed and electricity price is also almost linear if we consider changes with respect to the prices. However, in terms of wind speed, the values again reflect the nonlinear power curve model.

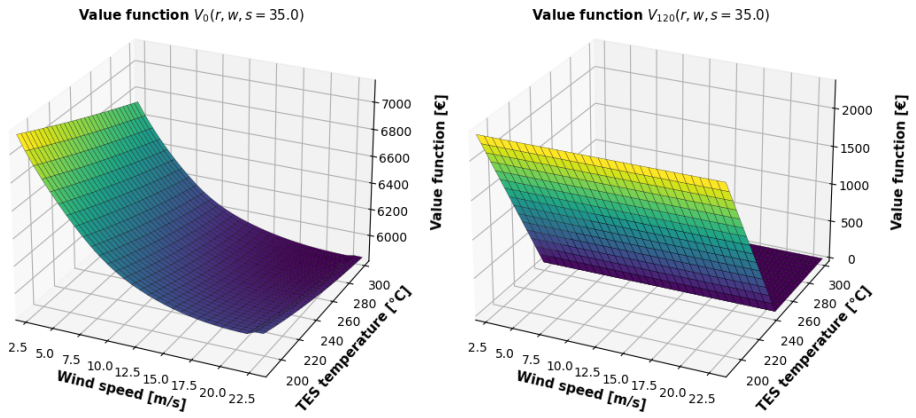


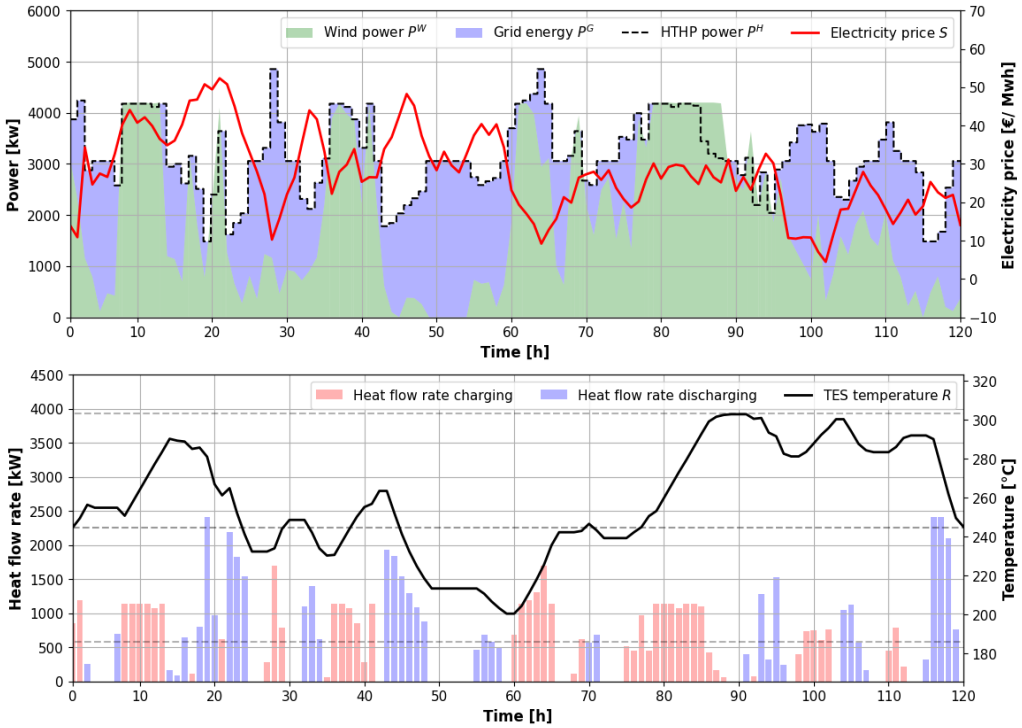
Figure 6. BDP: (**a**) Value function at initial time $n = 0$ (left) and at terminal time $n = 120$ (right) in terms of storage temperature and wind speed. (**b**) Value function at initial time $n = 0$ depending on the storage temperature and electricity price (left) as well as on wind speed and electricity price (right).

Second, we analyze a trajectory, starting from the initial state $(R_0, W_0, S_0) = (244.4, 4, 37)$ with $R_0 = r_{\text{crit}}$, for the control obtained by BDP as shown in Figure 7a. As expected, the control aims to charge the TES during periods of high wind power production and/or low prices. Charging that solely uses wind energy can be identified when the HTHP's electricity consumption (dashed black line) is covered by the available wind energy (green area), while additional power from the grid (blue area) is used to charge the TES when the price falls to small values.

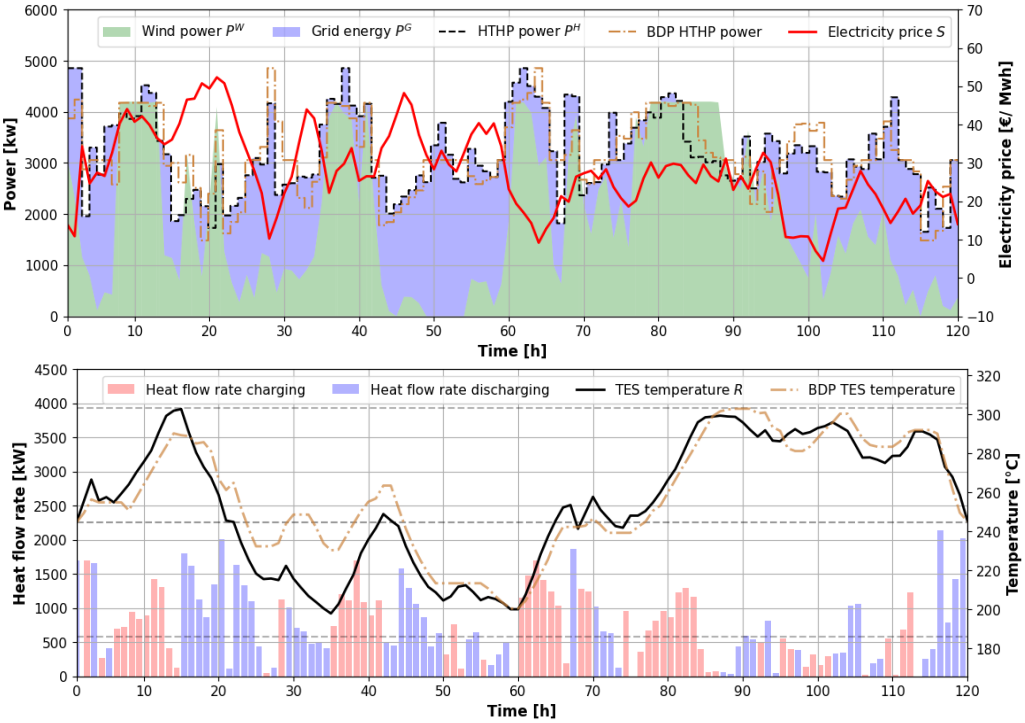
For instance, in hours 60 to 65, a combination of both scenarios can be observed. Conversely, when electricity prices are high and wind energy is not enough to cover the nominal HTHP power consumption, the TES is discharged, see, e.g., hours 43 to 48. Approximately four hours before the time horizon t_{E} is reached, the storage temperature is reduced to match the desired threshold r_{crit}.

For further details on the optimal control A, see Figure 10a, where the optimal decision rule is plotted as a function of time and TES temperature. At each time step $t_n = n\Delta t$, the decision rule is calculated for the corresponding TES temperature level (left y-axis) and the values of the seasonalities $\mu_W(t_n)$ and $\mu_S(t_n)$ at that time step. The seasonality functions μ_W (green line) and μ_S (dashed red line) with their corresponding scales are also presented (right y-axis). The colored red and blue areas correspond to the charging and discharging mode of the system with the respective heat flow rate. The white areas represent the idle mode of the system, i.e., no charging or discharging is operated.

It can be seen that the control obtained by BDP captures the functional structure of the seasonalities. This means that when prices are globally low (μ_S takes a global minimum), for example, at the hours 25, 50, or 75, charging is the preferred action. Conversely, discharging is performed when prices are high and wind speeds are low, see hours 20 or 45. For hours 15, 35, or 60, we can observe that when prices and wind speeds are locally high, the optimal action is to wait and operate in idle mode. Another situation where waiting is optimal appears when the storage temperature is higher than the critical temperature r_{crit} (dotted gray line) and prices are at the global minimum of μ_S, see, e.g., hour 45 to 55. However, when a maximum price is reached, the P2H system operates in discharge mode to compensate for the high electricity prices. Furthermore, when observing the seasonal patterns, it can be seen that the controls for charging, discharging, and idle mode are usually centered around the local and global extrema of μ_S. In particular, the electricity price appears to have a greater influence on the control than the wind speed (wind energy), as it has a major impact on the operational costs of the P2H system. An exception exists shortly before the terminal time t_{E}, when it is optimal to charge the TES if its temperature is below the critical temperature r_{crit}, regardless of the seasonal functions. This is to avoid penalties incurred at time t_{E} if the TES is not properly filled.



(**a**) BDP: Visualization of system dynamics and control for one trajectory.



(**b**) Q-learning: Visualization of system dynamics and control for one trajectory.

Figure 7. The upper plot in (**a**,**b**) shows the electricity consumption (dotted black) for operation the HTHP, with generated wind energy (green) and consumed grid power (blue) stacked, as well as the electricity price (red). The respective lower plots visualize the average TES temperature (black) and the transferred heat flow rate during charging (red) and discharging (blue). For a better comparison of both methods, we include the HTHP electricity consumption and TES temperature (brown) from the BDP solution in (**a**) into (**b**).

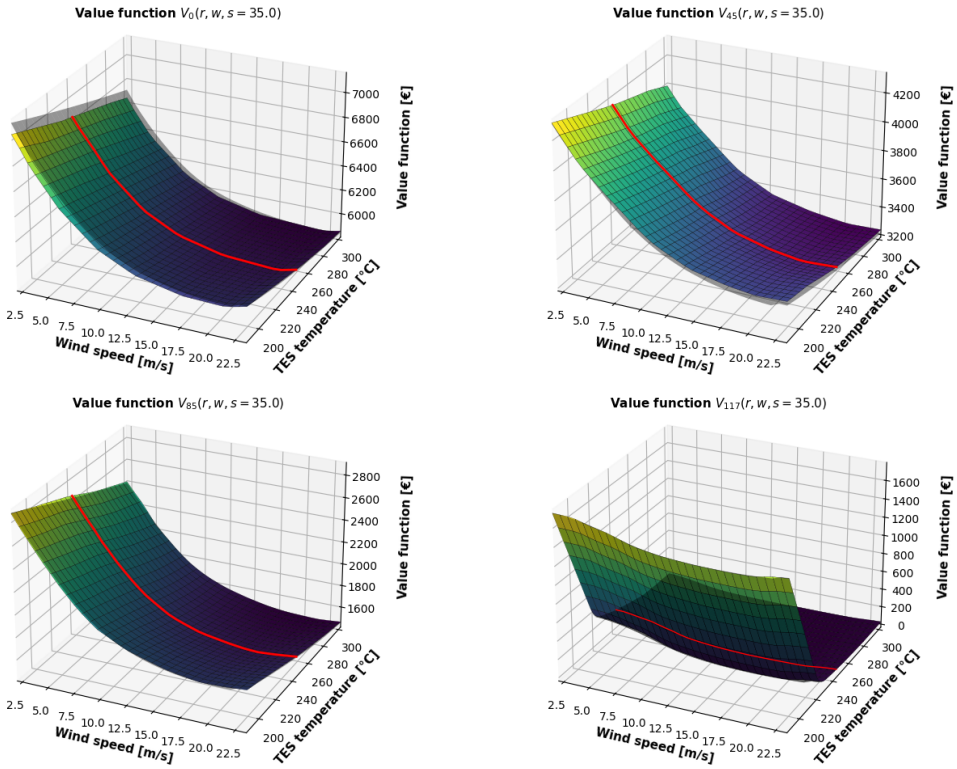
6.2. Q-Learning

We now compare the results of Q-learning using replay buffer, summarized in Algorithm 2, and BDP based on the computed value function and the corresponding control. Thus, BDP serves as a benchmark, as we can expect accurate results due to the high computational effort. For the parameterization of $\overline{Q}(x, a, \theta_n)$ a two-layer neural network is used with 128 neurons for each layer and ReLU activation functions. Since the parameterization is defined globally on the state space, no state discretization is required for BDP. The state distribution of the initial states $X_0 = x_0$ is chosen as a uniform random distribution on the discretized state space $\widetilde{\mathcal{X}}_0$. Figure 8a,b shows the value function of both methods depending on TES temperature and wind speed as well as on wind speed and electricity price, respectively, for the hours 0, 45, 85, and 117. More precisely, in each value function plot of Q-learning, the reference solution of BDP is visualized as a gray shaded graph. In addition, the red lines represent a cross-section of the value function, meaning that the value function is fixed in two variables and is visualized in order to provide a more detailed view at the error between both solutions. For a better comparison, the cross-sections (from Figure 8) are also depicted in Figure 9. Compared to BDP, Q-learning is able to capture the same shape of the value function. Especially for n near the terminal time, both function approximations differ only slightly. The further we move back in time from the terminal time, the more differences become visible.

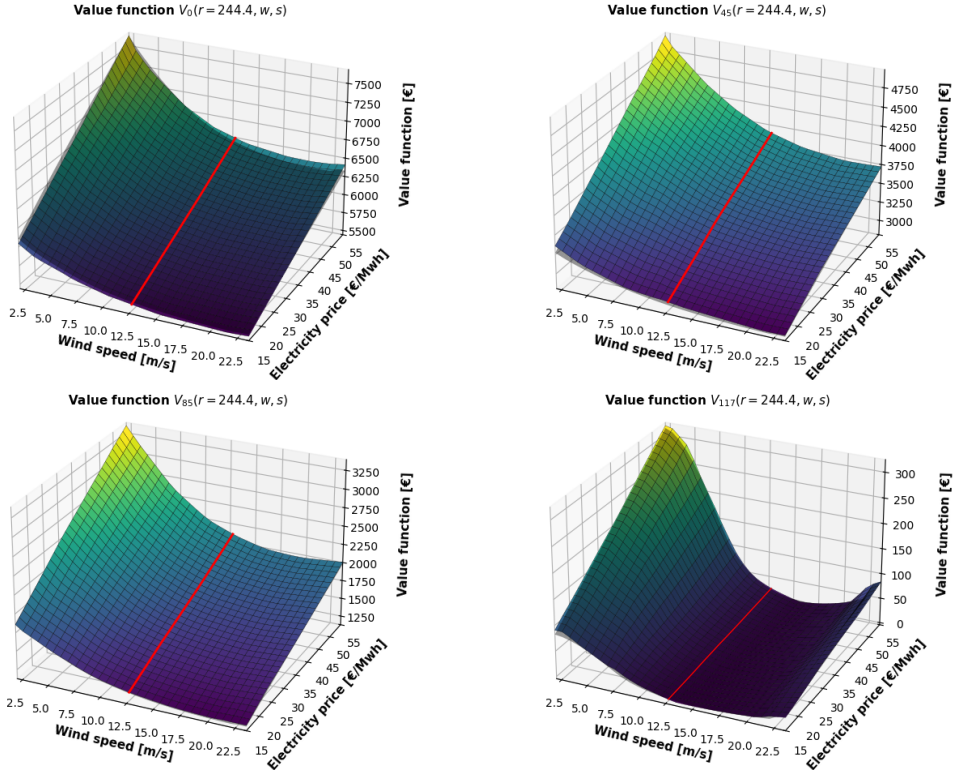
Figure 7b also confirms that the approximation of the value function by Q-learning is similar to that of BDP. The difference in the control is mainly reflected in the charging and discharging intensity of the TES and thus affects the HTHP's electricity consumption. Apart from this, the controller aims to charge the TES during times of high wind energy availability or/and low prices and to discharge vice versa. Overall, the comparison shows that the Q-learning control is qualitatively similar to that of BDP. However, an exception is the control in Figure 7b, which almost does not contain any waiting periods. Instead, the charging or discharging periods are generally longer compared to the BDP control, which can be seen in hours 15 to 25. Furthermore, Figure 10b provides a more detailed look at the optimal decision rule obtained with Q-learning. Obviously, the charging mode is performed in regions where price seasonality has a global minimum. However, almost every time a peak occurs, the high prices are compensated by discharging the TES. Again, the role of wind speed appears to be less important than the influence of the electricity price on the control itself. Even though seasonality is taken into account, the overall structure and sequence are not captured as well as with the BDP.

Computational Time.

In addition to the qualitative comparison of the numerical results with BDP and Q-learning, the computational effort required to compute the numerical solution is also of practical importance. Here, the computational time serves as an indicator of how well the methods are able to deal with the curse of dimensionality. All computations are performed on a compute server with 320 GB RAM running two Intel Xeon Gold 6136 processors, each with 12 cores and 24 threads. BDP calculates and saves the value function for the grid points of the discretized state spaces $\widetilde{\mathcal{X}}_n$. To speed up the computations of the value function in each time step, we will use all available cores and calculate its values for different grid points in parallel. In total, BDP requires 36 h computational time, which corresponds to approximately 18 min per time step. In contrast, Q-learning only takes around 8 h on a single core to compute the approximate solution. This means a time saving of a factor of 4 for the problem considered with three-dimensional states and one-dimensional actions. For stochastic optimal control problems in higher dimensions, we can expect much greater savings.



(**a**) Visualization of the value function in terms of storage temperature and wind speed.



(**b**) Visualization of the value function in terms of electricity price and wind speed.

Figure 8. Q-learning: Visualization of the value function at times $n = 0, 45, 85, 117$ depending on different state variables. The plot includes the BDP solution (gray) as a reference for comparison. The cross-sections (red) for each of the four value function plots are also shown in Figure 9 for better visualization.

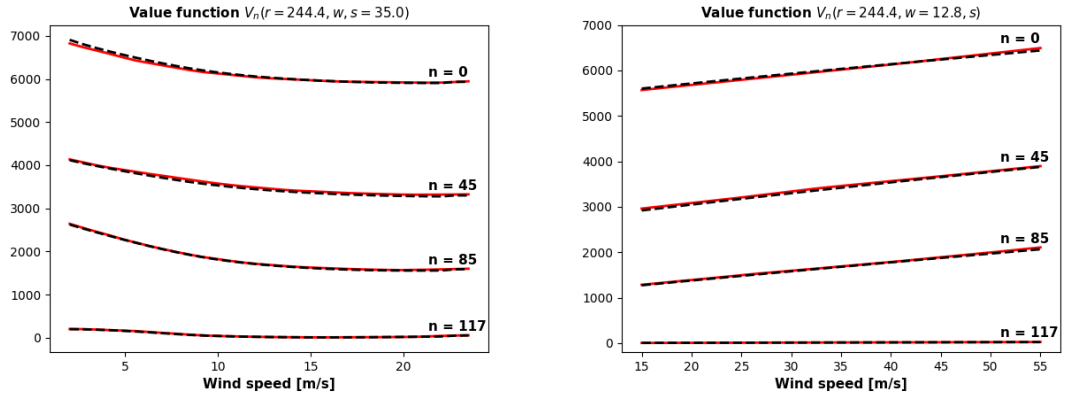
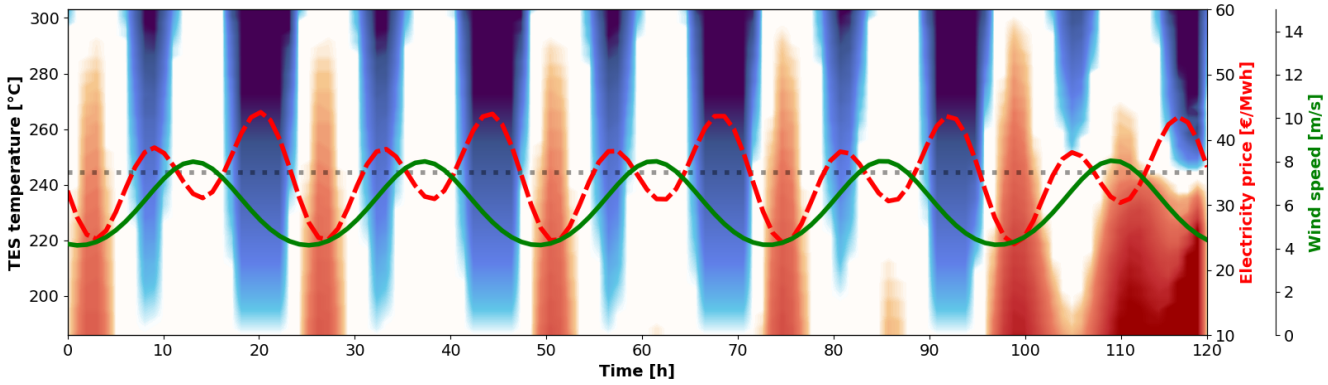
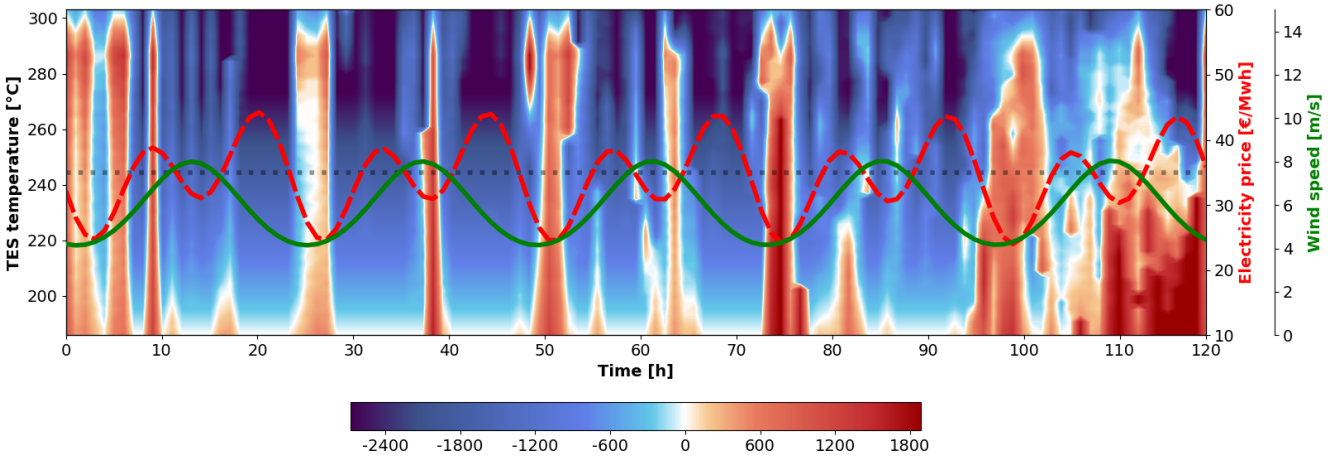


Figure 9. Visualization of the cross-sections from the value functions in Figure 8a (**left**) and Figure 8b (**right**). The black dashed curves show the value function approximation from BDP, which is compared with the value function from Q-learning given in red.



(**a**) Backward dynamic programming: Visualization of the decision rule u_n.



(**b**) Q-learning: Visualization of the decision rule u_n.

Figure 10. Visualization of the optimal decision rule u_n with respect to the TES temperature together with threshold temperature r_{crit} (dotted gray line) used in the terminal cost function (29). At each time point $t_n = n\Delta t$, the decisions calculated for the TES temperature is given in terms of the values of the seasonalities $\mu_W(t_n)$ (green line) and $\mu_S(t_n)$ (dashed red line), i.e., $u_n(r, w = \mu_W(t_n), s = \mu_S(t_n)) = a$. The light red and dark red colors represent low and high heat flow rates in charge mode, while light blue and dark blue represent low and high heat flow rates in discharge mode. White areas indicate the system's idle mode.

7. Conclusions

This work presents a mathematical model for the cost-optimal operation of an industrial P2H system. Apart from providing a modeling approach for the stochastic processes that takes into account correlation between wind speed and electricity price, we also calibrated the associated parameters with real-world data (see Supplementary Material, Section S6). The resulting discrete-time stochastic optimal control problem is formulated as an MDP and solved using the classical dynamic programming approach as well as modern reinforcement learning techniques, namely Q-learning.

A comparison of the numerical results shows that both methods can achieve similar approximations of the value function and yields reliable cost-optimal decision rules. Although the results of Q-learning differ in some aspects, it offers a faster and computationally more efficient solution for complex control problems. This is especially useful for problems with high-dimensional state and control spaces, where the dynamic programming approach will fail due to the curse of dimensionality.

By dropping the assumptions of constant mass flow $\dot{m}$ and waste heat temperature $T^{\mathrm{LT,in}}$, we can extend our model and make it more general. In this case, it is necessary to introduce $T^{\mathrm{LT,in}}$ as an additional state and $\dot{m}$ as an additional control variable, which increases the dimension of the control problem. Even though the classical backward recursion of dynamic programming might become intractable for this extended model due to the curse of dimensionality, we are confident that Q-learning still offers an efficient solution. However, as the dimension of the action space grows, it becomes infeasible to calculate the minimal action by discretization. Appropriate gradient descent methods could be used, at the cost that this may slow down the algorithm. Reinforcement learning algorithms such as policy gradients or actor-critic methods as in [27] might be more suitable for dealing with large (continuous) action spaces, as they already include a way to handle this minimization step.

Supplementary Materials: The following supporting information can be downloaded at: https://www.mdpi.com/article/10.3390/en19041046/s1. References [58–60] are cited in the Supplementary Materials.

Author Contributions: Conceptualization, methodology, software, formal analysis, writing—original draft preparation, and visualization, E.P. and M.B.; conceptualization, methodology, formal analysis, writing—review and editing, supervision, project administration, and funding acquisition, R.W. All authors have read and agreed to the published version of the manuscript.

Funding: E. Pilling and R. Wunderlich gratefully acknowledge the support by the Federal Ministry of Research, Technology and Space (BMFTR), award number 05M2022. We also acknowledge the support by the publication fund of BTU Cottbus-Senftenberg.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The code used to generate the results in this work is available from the corresponding author upon sincere request.

Acknowledgments: The authors thank Ibrahim Mbouandi Njiasse (BTU Cottbus-Senftenberg) for the valuable discussions that improved this paper.

Conflicts of Interest: The authors declare that they have no conflicts of interest to report for this paper.

Abbreviations and Symbols

The following abbreviations and symbols are used in this manuscript:

Acronyms

HTHP	High-temperature heat pump
HTF	Heat transfer fluid
TES	Thermal energy storage
LTHX	Low-temperature heat exchanger
HTHX	High-temperature heat exchanger
WT	Wind turbine
SG	Steam generator
P2H	Power-to-heat
MDP	Markov decision process
SDE	Stochastic differential equation
BDP	Backward dynamic programming
SGD	Stochastic gradient descent

Latin symbols

$T^{\mathrm{LT,in}}$	LTHX inlet temperature
$T^{\mathrm{SG,in}}$	SG inlet temperature
$T^{\mathrm{SG,out}}$	SG outlet temperature
$T^{\mathrm{HT,out}}$	HTHX outlet temperature
$T^{\mathrm{HT,in}}$	HTHX inlet temperature
T^{C}, T^{D}	TES outlet temp. charging/discharging
l^{C}, l^{D}	Charging/discharging factor
$\dot{m}$	Mass flow
m_{s}	Mass of TES
$c_{\mathrm{p,s}}$, $c_{\mathrm{p,f}}$	Heat capacity of TES/thermal oil
d	Rotational speed
P^{G}	Electrical power of grid
P^{H}	Electrical power of HTHP
P^{W}	Electrical power of WT
A	Heat flow rate
B	Brownian motion
X	State process
Y	Ornstein–Uhlenbeck process
R	Storage temperature
S	Electricity grid price
W	Wind speed
Z	Random disturbance
$\widehat{Z}$	Quantizer for Z
t	Time
t_{E}	Terminal time horizon
Δt	Time step size
n	Time point
N	Total number of time points
n_{H}	Number of HTHPs running in parallel
$\mathcal{X}$	State space
x	State variable
$\mathcal{A}$	Action space
$\mathcal{T}$	Transition operator
a	Control/action variable
C_n	Running cost
G	Terminal cost
J_n	Performance criterion
V_n	Value function
Q_n	State action function
θ	Parameter vector

Greek symbols

μ	Mean reversion level
λ	Mean reversion speed
σ	Volatility
η	Spread

References

1. Walden, J.V.M.; Bähr, M.; Glade, A.; Gollasch, J.; Tran, A.P.; Lorenz, T. Nonlinear operational optimization of an industrial power-to-heat system with a high temperature heat pump, a thermal energy storage and wind energy. *Appl. Energy* **2023**, *344*. [CrossRef]
2. Testi, D.; Urbanucci, L.; Giola, C.; Schito, E.; Conti, P. Stochastic optimal integration of decentralized heat pumps in a smart thermal and electric micro-grid. *Energy Convers. Manag.* **2020**, *210*, 112734. [CrossRef]
3. Kuang, J.; Zhang, C.; Sun, B. Stochastic dynamic solution for off-design operation optimization of combined cooling, heating, and power systems with energy storage. *Appl. Therm. Eng.* **2019**, *163*, 114356. [CrossRef]
4. Takam, P.H.; Wunderlich, R. Cost-optimal management of a residential heating system with a geothermal energy storage under uncertainty. *Int. J. Dyn. Control* **2025**, *13*, 424. [CrossRef]
5. Gu, W.; Wu, Z.; Bo, R.; Liu, W.; Zhou, G.; Chen, W.; Wu, Z. Modeling, planning and optimal energy management of combined cooling, heating and power microgrid: A review. *Int. J. Electr. Power Energy Syst.* **2014**, *54*, 26–37. [CrossRef]
6. Ehsan, A.; Yang, Q. Scenario-based investment planning of isolated multi-energy microgrids considering electricity, heating and cooling demand. *Appl. Energy* **2019**, *235*, 1277–1288. [CrossRef]
7. Zhong, J.; Tan, Y.; Li, Y.; Cao, Y.; Peng, Y.; Zeng, Z.; Nakanishi, Y.; Zhou, Y. Distributed Operation for Integrated Electricity and Heat System With Hybrid Stochastic/Robust Optimization. *Int. J. Electr. Power Energy Syst.* **2021**, *128*, 106680. [CrossRef]
8. Bui, V.H.; Hussain, A.; Kim, H.M. Q-Learning-Based Operation Strategy for Community Battery Energy Storage System (CBESS) in Microgrid System. *Energies* **2019**, *12*, 1789. [CrossRef]

9. Alabdullah, M.; Abido, M. Microgrid energy management using deep Q-network reinforcement learning. *Alex. Eng. J.* **2022**, *61*, 9069–9078. [CrossRef]
10. Watkins, C. Learning from Delayed Rewards. Ph.D. Thesis, King's College, Cambridge, UK, 1989.
11. Nakabi, T.; Toivanen, P. Deep reinforcement learning for energy management in a microgrid with flexible demand. *Sustain. Energy Grids Netw.* **2021**, *25*, 100413. [CrossRef]
12. Yu, L.; Xie, W.; Xie, D.; Zou, Y.; Zhang, D.; Sun, Z.; Zhang, L.; Zhang, Y.; Jiang, T. Deep Reinforcement Learning for Smart Home Energy Management. *IEEE Internet Things J.* **2020**, *7*, 2751–2762. [CrossRef]
13. Belloni, A.; Piroddi, L.; Prandini, M. A stochastic optimal control solution to the energy management of a microgrid with storage and renewables. In Proceedings of the 2016 American Control Conference (ACC), Boston, MA, USA, 6–8 July 2016; pp. 2340–2345. [CrossRef]
14. De Ridder, F.; Diehl, M.; Mulder, G.; Desmedt, J.; Van Bael, J. An optimal control algorithm for borehole thermal energy storage systems. *Energy Build.* **2011**, *43*, 2918–2925. [CrossRef]
15. Huang, C.; Seidel, S.; Jia, X.; Paschke, F.; Bräunig, J. Energy Optimal Control of a Multivalent Building Energy System using Machine Learning. In Proceedings of the 10th International Conference on Smart Cities and Green ICT Systems (SMARTGREENS 2021), Online, 28–30 April 2021; pp. 57–66. [CrossRef]
16. Fleming, W.H.; Soner, H.M. *Controlled Markov Processes and Viscosity Solutions*, 2nd ed.; Springer: New York, USA, 2006. [CrossRef]
17. Pham, H. *Continuous-Time Stochastic Control and Optimization with Financial Applications*, 1st ed.; Springer: Berlin/Heidelberg, Germany, 2009. [CrossRef]
18. Øksendal, B.; Sulem, A. *Applied Stochastic Control of Jump Diffusions*, 3rd ed.; Springer: Cham, Switzerland, 2019. [CrossRef]
19. Shardin, A.A.; Wunderlich, R. Partially observable stochastic optimal control problems for an energy storage. *Stochastics* **2017**, *89*, 280–310. [CrossRef]
20. Chen, Z.; Forsyth, P.A. Implications of a regime-switching model on natural gas storage valuation and optimal operation. *Quant. Financ.* **2010**, *10*, 159–176. [CrossRef]
21. Bäuerle, N.; Rieder, U. *Markov Decision Processes with Applications to Finance*, 1st ed.; Springer: Berlin/Heidelberg, Germany, 2011. [CrossRef]
22. Puterman, M.L. *Markov Decision Processes: Discrete Stochastic Dynamic Programming*; John Wiley & Sons: Hoboken, NJ, USA, 1994. [CrossRef]
23. Hernández-Lerma, O.; Lasserre, J.B. *Discrete-Time Markov Control Processes*, 1st ed.; Springer: New York, NY, USA, 1996. [CrossRef]
24. Powell, W.B. *Approximate Dynamic Programming: Solving the Curses of Dimensionality*; John Wiley & Sons: Hoboken, NJ, USA, 2011. [CrossRef]
25. Longstaff, F.A.; Schwartz, E.S. Valuing American Options by Simulation: A Simple Least-Squares Approach. *Rev. Financ. Stud.* **2001**, *14*, 113–147. [CrossRef]
26. Tsitsiklis, J.N.; Van Roy, B. Regression Methods for Pricing Complex American-Style Options. *IEEE Trans. Neural Netw.* **2001**, *12*, 694–703. [CrossRef]
27. Sutton, R.S.; Barto, A.G. *Reinforcement Learning: An Introduction*, 2nd ed.; MIT Press: Cambridge, MA, USA, 2018.
28. Pagès, G.; Pham, H.; Printems, J. An Optimal Markovian Quantization Algorithm for Multi-Dimensional Stochastic Control Problems. *Stochastics Dyn.* **2004**, *4*, 501–545. [CrossRef]
29. Li, X.; Verma, D.; Ruthotto, L. A Neural Network Approach for Stochastic Optimal Control. *SIAM J. Sci. Comput.* **2024**, *46*, C535–C556. [CrossRef]
30. Nielsen, M.A. *Neural Networks and Deep Learning*; Determination Press: San Francisco, CA, USA, 2015.
31. Goodfellow, I.; Bengio, Y.; Courville, A. *Deep Learning*; MIT Press: Cambridge, MA, USA, 2017.
32. Huré, C.; Pham, H.; Bachouch, A.; Langrené, N. Deep Neural Networks Algorithms for Stochastic Control Problems on Finite Horizon: Convergence Analysis. *SIAM J. Numer. Anal.* **2021**, *59*, 525–557. [CrossRef]
33. Takam, P.H.; Wunderlich, R.; Pamen, O.M. Modeling and simulation of the input–output behavior of ageothermal energy storage. *Math. Methods Appl. Sci.* **2023**, *47*, 371–396. [CrossRef]
34. Takam, P.H.; Wunderlich, R. Numerical Simulation of the Input-Output Behavior of a Geothermal Energy Storage. *Energies* **2025**, *18*, 1558. [CrossRef]
35. Wang, Y.; Hu, Q.; Li, L.; Foley, A.M.; Srinivasan, D. Approaches to wind power curve modeling: A review and discussion. *Renew. Sustain. Energy Rev.* **2019**, *116*, 109422. [CrossRef]
36. Bertsekas, D.P.; Tsitsiklis, J.N. *Neuro-Dynamic Programming*; Athena Scientific: Belmont, MA, USA, 1996.
37. Clifton, J.; Laber, E. Q-Learning: Theory and Applications. *Annu. Rev. Stat. Its Appl.* **2020**, *7*, 279–301. [CrossRef]
38. Pagès, G.; Printems, J. Optimal quadratic quantization for numerics: The Gaussian case. *Monte Carlo Methods Appl.* **2003**, *9*, 135–165. [CrossRef]
39. Pagès, G. A space quantization method for numerical integration. *J. Comput. Appl. Math.* **1997**, *89*, 1–38. [CrossRef]
40. Pagès, G. *Numerical Probability: An Introduction with Applications to Finance*, 1st ed.; Springer: Cham, Switzerland, 2018. [CrossRef]

41. Zador, P.L. Asymptotic Quantization Error of Continuous Signals and the Quantization Dimension. *IEEE Trans. Inf. Theory* **1982**, *28*, 139–149. [CrossRef]
42. Pagès, G. Introduction to Vector Quantization and Its Applications for Numerics. *ESAIM Proc. Surv.* **2015**, *48*, 29–79. [CrossRef]
43. Montes, T. Numerical Methods by Optimal Quantization in Finance. Doctoral Thesis, Sorbonne Université, Paris, France, 2020.
44. Lemaire, V.; Montes, T.; Pagès, G. New weak error bounds and expansions for optimal quantization. *J. Comput. Appl. Math.* **2020**, *371*, 112670. [CrossRef]
45. Konidaris, G.D.; Osentoski, S.; Thomas, P.S. Value Function Approximation in Reinforcement Learning Using the Fourier Basis. In Proceedings of the AAAI Conference on Artificial Intelligence, San Francisco, CA, USA, 7–11 August 2011; pp. 380–385. [CrossRef]
46. DeVore, R.; Hanin, B.; Petrova, G. Neural network approximation. *Acta Numer.* **2021**, *30*, 327–444. [CrossRef]
47. Elfwing, S.; Uchibe, E.; Doya, K. Sigmoid-weighted linear units for neural network function approximation in reinforcement learning. *Neural Netw.* **2018**, *107*, 3–11. [CrossRef]
48. Hornik, K.; Stinchcombe, M.; White, H. Multilayer Feedforward Networks Are Universal Approximators. *Neural Netw.* **1989**, *2*, 359–366. [CrossRef]
49. Robbins, H.; Monro, S. A Stochastic Approximation Method. *Ann. Math. Stat.* **1951**, *22*, 400–407. [CrossRef]
50. Garrigos, G.; Gower, R.M. Handbook of Convergence Theorems for (Stochastic) Gradient Methods. *arXiv* **2023**, arXiv:2301.11235. [CrossRef]
51. Melo, F.S.; Meyn, S.P.; Ribeiro, M.I. An analysis of reinforcement learning with function approximation. In Proceedings of the 25th International Conference on Machine Learning, Helsinki, Finland, 5–9 July 2008; Association for Computing Machinery: New York, NY, USA, 2008; pp. 664–671. [CrossRef]
52. Melo, F.S.; Ribeiro, M.I. Convergence of Q-learning with linear function approximation. In Proceedings of the 2007 European Control Conference (ECC), Kos, Greece, 2–5 July 2007; pp. 2671–2678. [CrossRef]
53. Cai, Q.; Yang, Z.; Lee, J.D.; Wang, Z. Neural Temporal-Difference Learning Converges to Global Optima. In Proceedings of the 33rd International Conference on Neural Information Processing Systems, NeurIPS, Vancouver, BC, Canada, 8–14 December 2019; Volume 32.
54. Xu, P.; Gu, Q. A Finite-Time Analysis of Q-Learning with Neural Network Function Approximation. In Proceedings of the 37th International Conference on Machine Learning, ICML, Vienna, Austria, 12–18 July, 2020; Volume 119, pp. 10555–10565.
55. De Asis, K.; Chan, A.; Pitis, S.; Sutton, R.S.; Graves, D. Fixed-Horizon Temporal Difference Methods for Stable Reinforcement Learning. *Proc. AAAI Conf. Artif. Intell.* **2020**, *34*, 3741–3748. [CrossRef]
56. Lin, L.J. Reinforcement Learning for Robots Using Neural Networks. Ph.D. Thesis, Carnegie Mellon University, Pittsburgh, PA, USA, 1992.
57. Mnih, V.; Kavukcuoglu, K.; Silver, D.; Graves, A.; Antonoglou, I.; Wierstra, D.; Riedmiller, M. Playing Atari with Deep Reinforcement Learning. *arXiv* **2013**, arXiv:1312.5602. [CrossRef]
58. Holý, V.; Tomanová, P. Estimation of Ornstein-Uhlenbeck Process Using Ultra-High-Frequency Data with Application to Intraday Pairs Trading Strategy. *arXiv* **2018**, arXiv:1811.09312. [CrossRef]
59. Quarteroni, A.; Sacco, R.; Saleri, F. *Numerical Mathematics*; Springer Science and Business Media: Berlin/Heidelberg, Germany, 2010.
60. Roussas, G. *An Introduction to Probability and Statistical Inference*; Academic Press: Cambridge, MA, USA, 2003.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.