Systems & Control Transactions

Research Article - Peer Reviewed Conference Proceeding
ESCAPE 35 - European Symposium on Computer Aided Process Engineering
Ghent, Belgium. 6-9 July 2025
Jan F.M. Van Impe, Grégoire Léonard, Satyajeet S. Bhonsale,
Monika E. Polańska, Filip Logist (Eds.)

PSE PRESS

Optimal Energy Scheduling for Battery and Hydrogen Storage Systems Using Reinforcement Learning

Moritz Zebenholzer[a]*, Lukas Kasper[a], Alexander Schirrer[b] and René Hofmann[a]

[a] TU Wien, Institute of Energy Systems and Thermodynamics, Vienna, Austria
[b] TU Wien, Institute of Mechanics and Mechatronics, Vienna, Austria
* Corresponding Author: moritz.zebenholzer@tuwien.ac.at

ABSTRACT

Optimal energy scheduling for sector-coupled multi-energy systems is becoming increasingly important as renewable energies such as wind and photovoltaics continue to expand. They are very volatile and difficult to predict. This creates a deviation between generation and demand that can be compensated for by energy storage technologies. For these, rule-based control is well established in industry, and mixed-integer model predictive control (MPC) is an area of research that promises the best results, usually regarding minimal costs. Drawbacks of MPC include the need for an adequate system model, often associated with high modeling effort, high computational effort for larger prediction horizons, and complications with stochastic variables. In this work, Reinforcement Learning is used in an attempt to overcome these difficulties without applying elaborate mixed-integer linear programming. The self-learning algorithm, which requires no explicit knowledge of the system behavior, can learn a control policy and uncertainties of the variables just by interaction with the (simulated) system model. It is demonstrated that Reinforcement Learning (exchange factor = 36.8 %) can learn complex system behavior with comparable quality to model predictive control (ex. = 32.4 %) and outperforms rule-based control (ex. = 41.8 %). This is done in a case study with the goal of minimizing the exchange of energy with the grid, with a battery and hydrogen system providing storage flexibility. These results were achieved in the context that the Reinforcement Learning agent only has instantaneous rather than predictive information, i.e., a very limited state of information compared to the MPC. The trained policy is then deployed while significantly decreasing the computational effort.

Keywords: Optimal Energy Scheduling, Reinforcement Learning (RL), Model-Predictive-Control (MPC)

INTRODUCTION

Due to the energy transition, the share of renewable sources of energy, such as wind and photovoltaic, is steadily increasing [1-3]. These are highly volatile, resulting in a discrepancy between generation and demand, which must be balanced by storage at any time but which is difficult to predict [2]. To accomplish this balancing efficiently and reliably, sector-coupled multi-energy systems (MES) combined with battery and hydrogen storage systems are deployed. These include batteries for daily fluctuations and electrolyzers, fuel cells and hydrogen storage systems for weekly volatility, for example. [1,2] As wind energy and photovoltaics, in particular, are decentralized and distributed, the electricity grid can be strained by uneven feed-in. More flexibility is therefore required at the place of power supply, whereby the energy exchange between production and load and the grid should be minimized. [1]

The optimal and safe operation of such MES requires operational planning, typically done by rule-based controllers (RBC) in industry [3]. In contrast, more elaborate model predictive control (MPC) strategies are the subject of current research. This form of optimal control is generally seen as delivering the best possible performance, usually aiming for minimum operating costs in compliance with the system-relevant constraints. [2,3]

However, the main obstacle to realizing MPC is that the optimization depends on an adequate model of the system dynamics. In addition, the uncertain prediction of

stochastic fluctuating quantities such as renewable energy generation, demand, and electricity prices significantly affect the control performance. Moreover, in scenarios that require long prediction horizons and detailed models, the arising mixed-integer optimization problems may require excessive computational effort. [2]

For these reasons, a control system that delivers optimal online performance without major computing effort is required. This can, for example, be achieved through Reinforcement Learning (RL), a form of machine learning that requires no explicit prior knowledge of the system dynamics. It can learn a control policy and the uncertainties of the input variables on the system only through repetitive interaction with the system model. RL has achieved excellent results in various disciplines, such as robotics and operational planning in industrial energy systems. [4]

Key contributions of this work are a consistent comparison between RBC, MPC and RL and an RL-based controller with comparable performance to MPC. It is demonstrated that this is possible with limited, instantaneous state information.

THEORETICAL BACKGROUND

The basic control methods of MPC [5] and RL [4], as well as their corresponding advantages and disadvantages with regard to industrial energy systems, are described below.

Model Predictive Control

In model predictive control, a mathematical model of the system dynamics called the design model hereafter is used to predict the system's future behavior based on the current state and a sequence of future control input values. The controlled system's response is optimized over a specific prediction horizon ΔT_{pred}. The schematic control loop is shown in Figure 1.

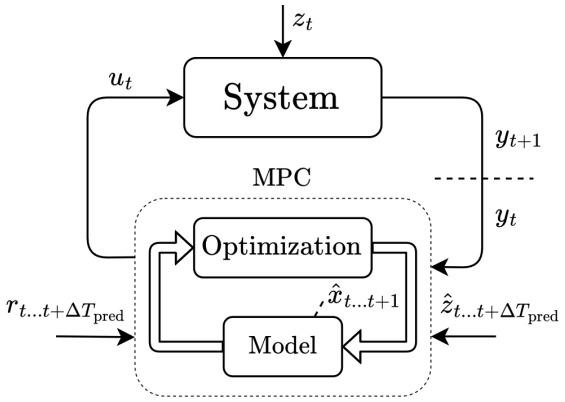

Figure 1. Model predictive control loop with control variables u_t, system state prediction $\hat{x}_{t\ldots t+T_{\text{pred}}}$ output y_t, disturbance z_t and reference r_t for timestep t and prediction horizon ΔT_{pred} adapted from [5].

For simplicity of time discrete notation, index t refers to the current sample, $t+1$ to the next, and $t+\Delta T_{\text{pred}}$ to the last sample in the current prediction horizon where the full trajectory $o_{t\ldots t+\Delta T_{\text{pred}}}$ is denoted hereafter with only the basis o. The sequence of the control variables u, is determined so that a predefined objective J, a cost function, is minimized, see Equation 1. These can be, for example, minimal costs, minimal deviations from a reference trajectory r or minimal energy consumption. Constraints are imposed on the system's input variables u and design model state estimates $\hat{x}$ via equality $(h(\cdot)=0)$ and inequality $(g(\cdot)\leq 0)$ constraints to ensure safe and physically feasible operation. The set X contains both real and integer-valued variables.

$$\begin{aligned} &\underset{u}{\operatorname{argmin}} \;\; J(\hat{x},u,\hat{z},r) \\ &\text{s.t.} \quad h(x,u)=0 \\ &g(x,u)\leq 0 \\ &x\in X, u\in U \end{aligned} \tag{1}$$

Only the control variables u_t of the first-time step t are applied to the system; all others are discarded. At the next time step, $t+1$, this iterative optimization process is repeated, using new predictions and measured values of the output y_{t+1} and associated, updated system state estimate $\hat{x}_{t+1}$, also known as *receding horizon control*. It is particularly suitable for multivariable systems that contain complex relationships, relevant constraints and well-predictable inputs.

When both continuous and discrete variables are involved, as is often the case with energy systems, the optimal control problem (1) is usually formulated as a mixed-integer linear program (MILP) and solved using branch and bound algorithms [12]. Since the variables are typically linked across several time steps and constraints, it is difficult to decompose the problem. Binary variable formulations lead to combinatorial complexity, which becomes intractable as problem size increases. However, for practical applications, a large prediction horizon is often required, increasing the number of decision variables with each discrete time step and, therefore, the computational effort significantly. Although the MILP problem can be solved, it may be too complex for the necessary sampling time step size.

Reinforcement Learning

In Reinforcement Learning, an entity called *agent* interacts with its environment with the purpose of learning to make sequential decisions. It receives a state observation s_t and prior reward r_t signal and performs an action a_t that influences the environment, which evolves according to the internal system dynamics. The successor state s_{t+1} and reward r_{t+1} are the result. This process is sketched in Figure 2 and is understood in a stochastic sense as a Markov Decision Process (MDP).

Here, the agent learns by trial and error, guided by a reward function. Its goal is to maximize the discounted, cumulative reward over the considered period (episode)

$$\max \sum_{t=1}^{T} r(s_t, a_t, s_{t+1}) \cdot \gamma^{t-1}, \qquad (2)$$

with discount factor γ. While deriving a control law $a_t = \pi(s_t)$, which maps states to actions. Striking a good balance between exploration, i.e., visiting new state-action combinations, and exploitation, i.e., taking the actions with the highest cumulative reward expectation, is essential to the training process.

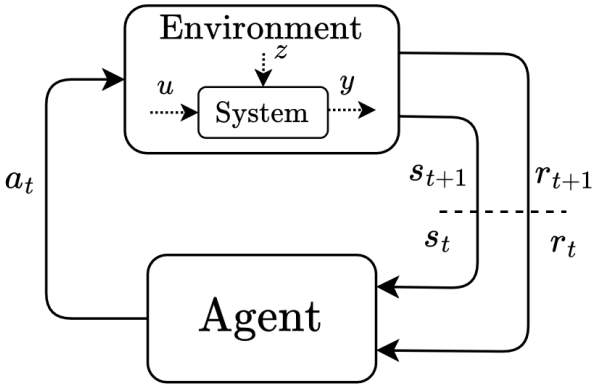

Figure 2. Reinforcement Learning control loop with control variables u, output y and disturbance z, as well as, state observation s_t, action a_t and reward r_t for timestep t adapted from [4].

In deep RL, the internal relationships in the agent are represented by deep neural networks (NN), which endows it with the ability to model any non-linear, high-dimensional function. Another advantage is that RL does not depend on modeling using first principles but learns from direct experience, allowing it to adapt to changing system dynamics even after the training process. This allows the performance to improve over time and can help reduce the prediction's uncertainty.

However, this method has difficulties with performance and stability guarantees, as it is challenging to provide for neural networks. In addition, the learning process is very sample-inefficient and a convergent learning process is not guaranteed in general. These topics are highly active fields of research. [6]

METHODS

This chapter first outlines the use case and the underlying modeling adapted from [3]. It then links to the control methods and discusses the parameterization and implementation.

Use Case

RL's potential is demonstrated through a representative use case, a MES providing storage flexibility. It consists of a battery (BSS) and a hydrogen storage system (HSS) comprising an electrolyzer, a fuel cell and a hydrogen storage unit. This MES, shown in Figure 3, is influenced, on the one hand, by generation p_{gen} and consumption p_{de} and, on the other hand, by the energy exchange not compensated for by the storage systems with the grid ($p_{\text{gr}}^{\text{in}}$,$p_{\text{gr}}^{\text{out}}$). All quantities are non-negative.

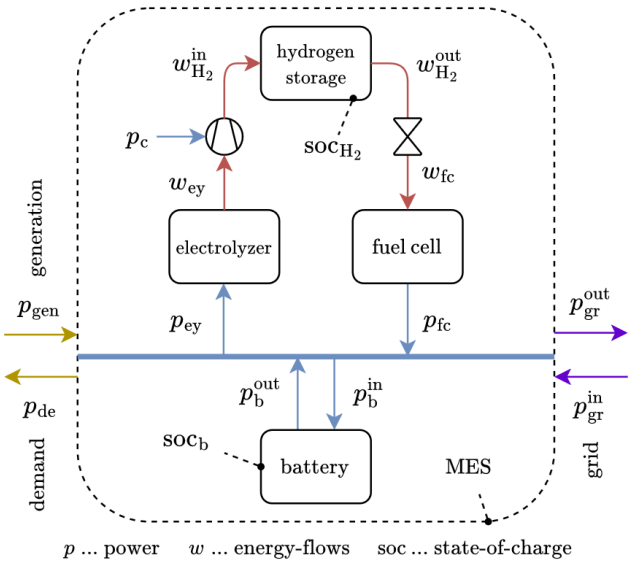

Figure 3. Multi-energy system (MES) with battery (b) and hydrogen storage systems, comprising electrolyzer (ey), storage (H_2) and fuel cell (fc).

The components can be modeled using the following relationships for the battery

$$\text{soc}_{\text{b,t+1}} = \text{soc}_{\text{b,t}} + \frac{\Delta t_{\text{sim}}}{C_{\text{b}}} \begin{cases} p_{\text{b},t}^{\text{in}} \cdot \eta_{\text{b}}, \text{charge} \\ -p_{\text{b},t}^{\text{out}} / \eta_{\text{b}}, \text{ discharge} \end{cases} \qquad (3)$$

where soc_{b} is the state-of-charge, η the efficiency and C_{b} the battery's capacity. The battery can either be charged by p_{b}^{in} or discharged by $p_{\text{b}}^{\text{out}}$. The hydrogen storage is modeled using

$$\text{soc}_{\text{H}_2,\text{t+1}} = \text{soc}_{\text{H}_2,\text{t}} + \frac{\Delta t_{\text{sim}} \cdot \eta_{\text{H}_2}}{C_{\text{H}_2}} \left(w_{\text{H}_2,t}^{\text{in}} - w_{\text{H}_2,t}^{out} \right), \qquad (4)$$

and analogously soc_{H_2} is the state-of-charge and C_{H_2} is the capacity of the hydrogen storage. The energy level is increased when hydrogen is fed in ($w_{\text{H}_2}^{in}$) and decreased by $w_{\text{H}_2}^{\text{out}}$. The electrolyzer and the fuel cell are modeled using non-linear equations $w_{\text{ey}} = f_{\text{ey}}(p_{\text{ey}})$ and $w_{\text{fc}} = f_{\text{fc}}(p_{\text{fc}})$, respectively. An isentropic compressor, used to increase the pressure pr, using power $p_{\text{c}} = f_{\text{c}}(w_{\text{ey}}, pr)$, is located upstream of the storage tank, whereby the following applies, $w_{\text{H}_2}^{in} = w_{\text{ey}}$. Downstream of the hydrogen storage there is a valve for which $w_{\text{H}_2}^{\text{out}} = w_{\text{fc}}$ applies. For the MPC

design model all non-linear equations are piecewise linearized.

The energy equation

$$p_{\text{gen}} - p_{\text{de}} + p_{\text{fc}} - p_{\text{ey}} + p_{\text{b}}^{\text{out}} - p_{\text{b}}^{\text{in}} - p_{\text{c}} + p_{\text{gr}}^{\text{in}} - p_{\text{gr}}^{\text{out}} = 0 \quad (5)$$

links the various components' powers. The performance indicator, the exchange factor ϵ_t, is defined as the ratio between the energy exchange with the grid and the difference of generation and demand sampled with Δt_{sim}

$$\epsilon_t = \frac{\sum_{\tau=0}^{\tau=t} p_{\text{gr},\tau}^{\text{in}} + p_{\text{gr},\tau}^{\text{out}}}{\sum_{\tau=0}^{\tau=t} |p_{\text{gen},\tau} - p_{\text{de},\tau}|}. \quad (6)$$

Here, $\epsilon_t = 0$ means that storage flexibility can compensate for any difference between generation and consumption. If $\epsilon_t = 1$, the entire difference is passed through to the grid.

Control Engineering

In the control engineering sense, the process variables for the MPC can be divided up as follows. The control variable u_t consists of the power flows into and out of the battery, the electrolyzer and the fuel cell

$$u_t = \left[p_{\text{b}}^{\text{in}} \quad p_{\text{b}}^{\text{out}} \quad p_{\text{ey}} \quad p_{\text{fc}}\right]_t^T. \quad (7)$$

The system's output y_t comprises the state-of-charge for battery and hydrogen storage and the power exchange with the grid:

$$y_t = \left[\text{soc}_{\text{b}} \quad \text{soc}_{\text{H}_2} \quad p_{\text{gr}}^{\text{in}} \quad p_{\text{gr}}^{\text{out}}\right]_t^T. \quad (8)$$

The disturbance forecast $\hat{z}$, sampled with Δt_{opt}, consists of the generation and demand prediction from timestep t to the prognosis horizon $t + \Delta T_{\text{pred}}$ acting on the system

$$\hat{z} = \begin{bmatrix} p_{\text{gen},t} & p_{\text{de},t} \\ \vdots & \vdots \\ p_{\text{gen},t+\Delta T_{\text{pred}}} & p_{\text{de},t+\Delta T_{\text{pred}}} \end{bmatrix}^T. \quad (9)$$

The objective of the MPC is to minimize the exchange of energy with the grid, i.e., the area under the absolute value of the power curve using a sampling time of Δt_{opt},

$$\underset{u}{\text{argmin}} \sum_t^{t+\Delta T_{\text{pred}}} \left(p_{\text{gr},t}^{\text{in}} + p_{\text{gr},t}^{\text{out}}\right) \cdot \Delta t_{\text{opt}}. \quad (10)$$

The action a_t of the RL agent is equivalent to the control variable u_t. The observation state s_t consists in part of the system output y_t and the disturbance $\hat{z}_t$ and can be written as

$$s_t = \left[\text{soc}_{\text{b}} \quad \text{soc}_{\text{H}_2} \quad p_{\text{gen}} \quad p_{\text{de}}\right]_t^T \quad (11)$$

and reward information comprises

$$r_t = \left[p_{\text{gr}}^{\text{in}} \quad p_{\text{gr}}^{\text{out}}\right]_t^T. \quad (12)$$

The reward function of the RL agent, sampled with Δt_{opt}, and chosen discount factor as $\gamma = 0.99$ can be written as

$$\max \sum_t^{t+\Delta T_{\text{pred}}} -\left(p_{\text{gr},t}^{\text{in}} + p_{\text{gr},t}^{\text{out}}\right) \cdot \Delta t_{\text{opt}} \cdot \gamma^{t-1}, \quad (13)$$

and a sampling time of Δt_{opt} is used. All system-specific constraints are represented in the design model for the MPC and inherently in the simulation model for RL training and for the method comparison.

Parameters & Data

Generation and demand data for Austria for 2023 were used as the data basis [7]. Five representative weeks according to their distribution were calculated, and one exemplary week (Figure 4, 19.6 - 26.6.2023) was used to test the use case. Energy production comprises 75 % photovoltaics and 25 % wind power. The data was normalized and scaled so that, on average, the demand is $\overline{P_{\text{de}}} = 250\,kW$ and generation $\overline{P_{\text{gen}}} = 300\,kW$, respectively. The data was subjected to white noise, whereby the signal-to-noise ratio is a factor of 20. The non-noisy data is used for the prediction of the MPC and for training the RL agent, while the noisy curves are declared as ground truth in the simulation.

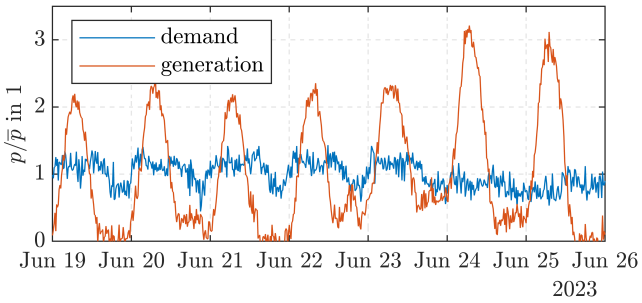

Figure 4. Demand and generation for a representative summer week of Austria, 2023 [7].

The MES components' parameters are listed in Table 1, and the simulation data in Table 2.

Table 1. MES Properties

Battery System		Hydrogen System		
Property	Value	Property	Value	Unit
C_{b}	500	C_{H_2}	3000	kWh
p_{b}^{max}	100	p_{ey}^{max}	400	kW
		p_{fc}^{max}	100	kW
$\text{soc}_{\text{b}}^{min}$	0.2	$\text{soc}_{\text{H}_2}^{min}$	0.1	-
$\text{soc}_{\text{b}}^{max}$	0.8	$\text{soc}_{\text{H}_2}^{max}$	0.9	-
η_{b}	0.9	η_{H_2}	1	-

Simulation & Training

In the simulation environment, the system equations are solved with a time step size of Δt_{sim}. After Δt_{opt}, either the optimization for the MPC is solved or the trained neural network for RL is evaluated. For the MPC, the

prediction horizon ΔT_{pred} is divided into 96 time-steps with size Δt_{opt}. The RL agent is trained with a sampling time of Δt_{opt}. Proximal Policy Optimization (PPO) [8] is used as the RL algorithm, where it is trained for ~10^4 episodes.

Table 2. Simulation Properties

Property	Value	Unit
Δt_{sim}	1	min
Δt_{opt}	15	min
ΔT_{pred}	24	h
ΔT_{sim}	168	h

The neural network architecture used for the value-function critic is a fully connected, feedforward NN with ~1200 degrees of freedom. A stochastic Gaussian actor is implemented with ~3500 degrees of freedom.

Implementation

The simulation environment was programmed in MatLab [9], whereby the optimization problem (MPC) was set up with YALMIP [10] and solved with Gurobi [11]. The PPO agent was trained with the MatLab Reinforcement Learning Toolbox [12] and 64 parallel workers. These were simulated on a system with 128 cores and 256 GB RAM (AMD EPYC 7702P).

RESULTS & DISCUSSION

The results of the simulation are the process variables of the use case, whereby the control variables resulting from the optimization (MPC, NN-RL) and the state-of-charges are shown in Figure 6. Two tiles are used for the MPC (top) and two for the RL (bottom). These, in turn, are divided into hydrogen-related variables of the electrolyzer p_{ey}, the fuel cell p_{fc} and the storage tank p_{H_2} as well as battery-related variables charge p_{b}^{in}, discharge $p_{\text{b}}^{\text{out}}$ and state of charge soc_{b}. The powers are normalized, i.e. divided by the maximum power, because of $p_{\text{min}} = 0$. Without loss of relevant information aspects, three days were chosen from the test week and

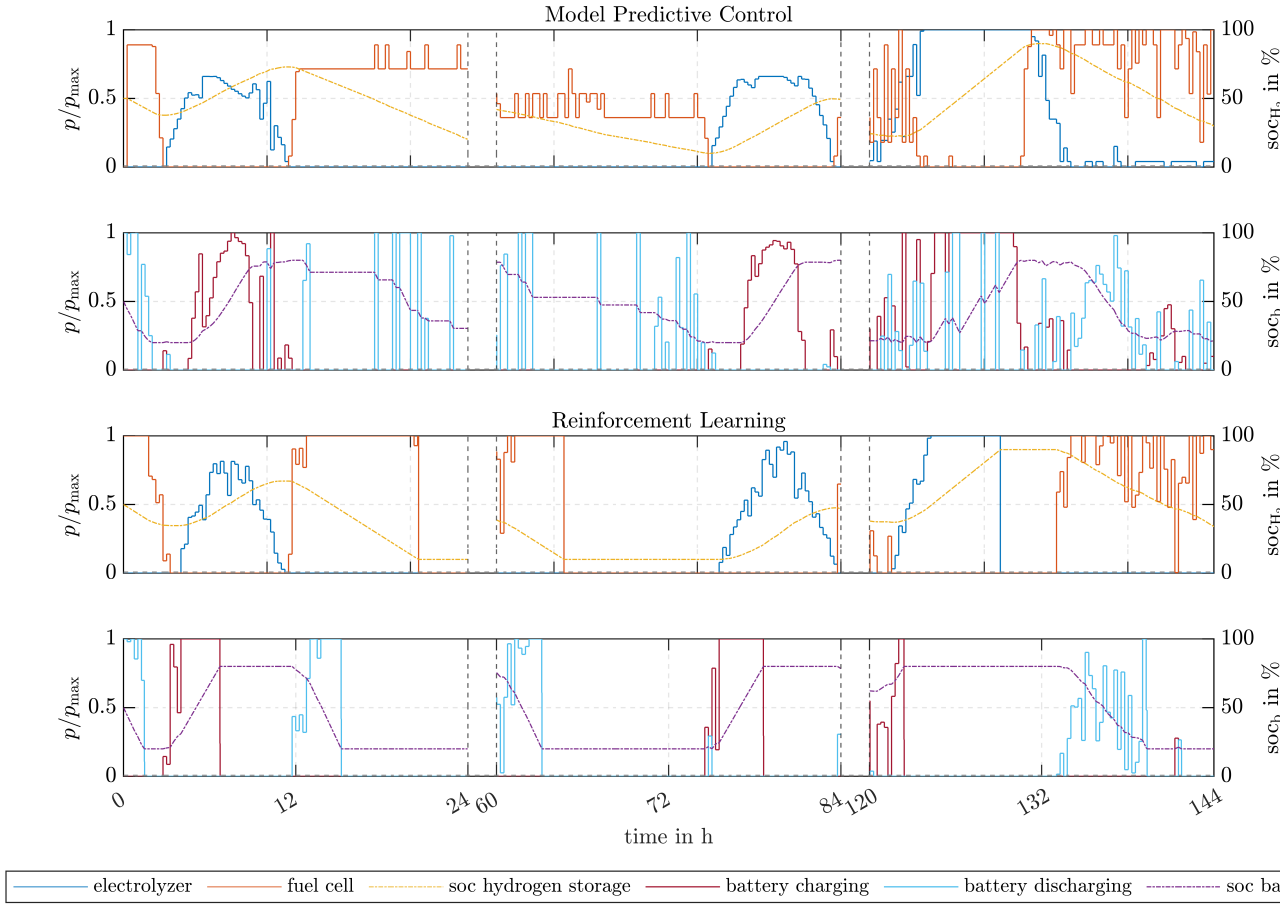

Figure 5. Setpoints electrolyzer power p_{ey}, fuel cell power p_{fc} and fill level of storage tank soc_{H_2} as well as battery-related variables charging power p_{b}^{in}, discharging power $p_{\text{b}}^{\text{out}}$and state-of-charge soc_{b} for multi-energy system use case for model predictive control and Reinforcement Learning.

displayed for better visualization. In principle, as can be seen in Figure 5, two different operating modes can be identified: If more energy is generated than consumed, battery charging and electrolyzer are activated, and if there is a negative residual load, the battery is discharged, and the fuel cell is used. This only works as long as the storage units are not at their limits. Therefore, Predictive information is required on how the externally imposed energy flows will adjust to provide optimal flexibility. However, only the MPC has access to the inherent forecast information $\hat{z}_{t\ldots t+\Delta T_{\text{pred}}}$. The RL agent, which only has access to instantaneous observations, can only learn an implicit average prediction for future states through the sequence of observed states (trajectory). However, this assumes a similarity between the trained and tested generation and demand curves, which has been fulfilled in the tested scenario. Figure 6 shows the exchange factor ε_t for MPC and RL, whereby the first 24 h can be regarded as a run-in effect, as the MPC has a forecast horizon of 1 day. The performance is comparable in the first 5 days; a difference can only be seen when a lot of renewable energy is produced and little is consumed.

In these situations, it is optimal to use a discontinuous switching behavior of the storage. It can be concluded that the lack of predictive information hinders the RL agent from performing better. After a one-week test phase, there is a difference in exchange factor of approx. 4 %.

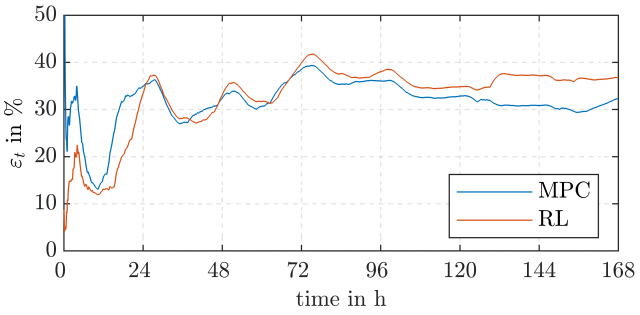

Figure 6. Energy exchange factor ε_t with grid for 1 week.

CONCLUSION & OUTLOOK

In this study, the performance of a Reinforcement Learning-based control strategy for sector-coupled multi-energy systems providing storage flexibility was evaluated. With Reinforcement Learning, a performance comparable to that of model predictive control can be achieved, which can be regarded as the upper limit in the nominal case. The Reinforcement Learning agent can outperform the industry standard rule-based control. With only instantaneous observations and without complex mixed-integer linear program modeling, the system can be controlled by a self-learning algorithm. The following questions are to be answered in a further study:

Can the Reinforcement Learning agent achieve better control quality if the same predictive information is provided for complete comparability? Can the self-learning system accomplish the same performance as model predictive control if unpredictable errors can be learned better than represented by a deterministic MPC?

ACKNOWLEDGEMENTS

The authors want to acknowledge the support provided by the doctoral school *Smart Industrial Concept* (https://www.tuwien.at/doc/sic) and the TU Wien Bibliothek for financial support through its Open Access Funding Programme.

REFERENCES

1. Le, T. S., et al. (2023). Optimal sizing of renewable energy storage: A techno-economic analysis of hydrogen, battery and hybrid systems considering degradation and seasonal storage. Applied Energy, 336, 120817. https://doi.org/10.1016/j.apenergy.2023.120817
2. Van, L. P., et al. (2023). Review of hydrogen technologies based microgrid: Energy management systems, challenges and future recommendations. International Journal of Hydrogen Energy, 48(38), 14127–14148. https://doi.org/10.1016/j.ijhydene.2022.12.345
3. Holtwerth, A., et al. (2024). Closed-loop model predictive control of a hybrid battery-hydrogen energy storage system using mixed-integer linear programming. Energy Conversion and Management: X, 22, 100561. https://doi.org/10.1016/j.ecmx.2024.100561
4. Perera, A. T. D., & Kamalaruban, P. (2021). Applications of reinforcement learning in energy systems. Renewable and Sustainable Energy Reviews, 137, 110618. https://doi.org/10.1016/j.rser.2020.110618
5. Schwenzer, M., et al. (2021). Review on model predictive control: An engineering perspective. The International Journal of Advanced Manufacturing Technology, 117(5–6), 1327–1349. https://doi.org/10.1007/s00170-021-07682-3
6. Jin, M., & Lavaei, J. (2020). Stability-Certified Reinforcement Learning: A Control-Theoretic Perspective. IEEE Access, 8, 229086–229100. https://doi.org/10.1109/ACCESS.2020.3045114
7. ENTSO-E Transparency Platform. (n.d.). Retrieved January 14, 2025, from https://transparency.entsoe.eu
8. Schulman, J., et al. (2017). Proximal Policy Optimization Algorithms (arXiv:1707.06347). arXiv. https://doi.org/10.48550/arXiv.1707.06347

9. The MathWorks Inc. (2024). MATLAB version: 23.2 (R2023b), Natick, Massachusetts: The MathWorks Inc. https://www.mathworks.com.
10. Löfberg, Johan. “A toolbox for modeling and optimization in MATLAB.” Proceedings of the CACSD Conference (2004), https://www.mathworks.com.
11. The MathWorks Inc. (2024). Reinforcement Learning Toolbox version: 23.2 (R2023b), Natick, Massachusetts: The MathWorks Inc. https://www.mathworks.com.
12. Gurobi Optimization, LLC. “Gurobi Optimizer Reference Manual.” (2024). https://www.gurobi.com.