Safe and Efficient Control of a Brayton Cycle Heat Pump Using Reinforcement Learning

A. Phong Tran[1] Fatma Cansu Yücel[1]

[1]German Aerospace Center (DLR), Institute for Low-Carbon Industrial Processes, Germany,
{Phong.Tran,Fatma.Yuecel}@dlr.de

Abstract

Decarbonizing industrial process heating will increasingly depend on high-temperature heat pumps. In particular, Brayton cycle heat pumps, which can reach temperatures above 250 °C, are viewed as a promising technology. However, ensuring safe operation and optimal control remains challenging. This study presents an experimentally validated dynamic model of a Brayton cycle heat pump, a system with multiple control inputs for regulating its thermal output. Using this model as a training environment, several control concepts integrating Reinforcement Learning (RL) and traditional PI controllers were implemented to achieve desired heat supply at target temperatures. Domain randomization was employed to improve the controller robustness against model uncertainties in preparation for deployment on the physical system. The results demonstrate that RL controllers can not only achieve the desired set-point temperature under varying loads while maintaining required safety margins, but also discovered a novel, more energy-efficient operational strategy.

Keywords: Brayton cycle heat pump, dynamic simulation, reinforcement learning, model-based control design

1 Introduction

Heating for industrial processes is predominantly fueled by fossil fuels, presenting a significant challenge for decarbonization efforts (IEA 2024). High-temperature heat pumps are crucial technologies for electrifying industrial process heating, and improving their control is essential for maximizing their operational efficiency and decarbonization impact.

1.1 High-temperature heat pumps

Recent advancements have significantly raised achievable supply temperatures of heat pump systems. The IEA HPT Annex 58 (Heat Pump Centre 2023) compiled heat pump demonstrators achieving temperatures of 115 °C to 280 °C. For the upper end of this range, particularly for sensible heating, gas compression heat pumps utilizing the Brayton cycle are emerging as a viable technology, with studies indicating technical feasibility up to 300 °C to 400 °C and economic potential in scenarios reaching 280 °C (Zühlsdorf et al. 2019). Oehler, Tran, and Stathopoulos (2022) note that while Brayton cycle heat pumps provide significant operational flexibility, their operation is constrained by compressor surge, thermal stresses, and natural frequencies, thus requiring a carefully designed control system.

1.2 Reinforcement learning

Reinforcement Learning (RL) is an area of machine learning focused on how agents learn to make optimal sequences of decisions through interaction with an environment. RL has proven useful in many areas, including games (Silver et al. 2018), robotic systems (Tang et al. 2025), and optimal energy dispatching (Di Cao et al. 2020). Model-free RL algorithms are particularly adaptable as they learn purely from environment interactions, but this often requires large amounts of data, highlighting the need for simulation. Modelica models have emerged as suitable simulators to train RL algorithms, as demonstrated by developments such as FMUGym (Wrede et al. 2024) and ModelicaGym (Lukianykhin and Bogodorova 2019), both of which utilize the Functional Mock-Up Interface (FMI). Although training in simulated environments offers a safe and efficient approach, these simulations inherently differ from their real-world counterparts. This discrepancy can cause policies trained in simulation to fail or underperform when deployed on the actual physical system. This challenge is widely recognized as the *reality gap* or the *Sim2Real* problem (Zhao, Queralta, and Westerlund 2020).

1.3 Other advanced control techniques

Optimica (Åkesson 2008), an extension of the Modelica language, allows users to formulate optimization problems alongside dynamic system models, which can then be processed by tools such as JModelica (Magnusson and Åkesson 2015) and the Optimica Compiler Toolkit (Modelon, Inc. 2024). These tools interface with the framework CasADi (Andersson et al. 2019), which translates the problem into a non-linear programming problem. The resulting NLP can then be solved by numerical solvers like Interior Point Optimizer (IPOPT). The described toolchain provides a mechanism for solving optimal control problems for systems modeled in Modelica. This capability is a prerequisite for Model Predictive Control (MPC).

1.4 Our contributions

This study utilizes an experimentally validated dynamic model of a Brayton cycle heat pump as an accurate training environment for RL. Three control concepts are comparatively analyzed, including a traditional approach with PI controllers, a pure RL controller and a combination of RL and PI controllers. To improve the potential for real-world transfer, the RL agents are trained using domain randomization. The results show that RL agents can learn the control tasks and discover novel and more energy-efficient operational strategies. The viability of training RL agents in a simulation environment is demonstrated by their control performance and ability to generalize across a wide range of system parameters.

2 The CoBra Heat Pump

This study investigates a closed-loop Brayton-cycle heat pump that uses air as its working fluid. The prototype under investigation, named "CoBra", was developed in Cottbus, Germany, and serves as a research platform for the design and operation of high-temperature heat pumps. Yücel et al. (2025) provide details on the system design and first commissioning tests of the plant.

2.1 System description

Figure 1 shows a flow schematic with the main components of the heat pump. The heat pump employs a two-stage compression system (A), using two radial compressors connected in series driven by an electric motor. After compression, the working fluid releases heat in the high-temperature heat exchanger (B). A three-way valve (C) controls the flow through the recuperator (G), which preheats the flow before compression using the temperature difference between the states 2 and 5. The turbine (D) expands the fluid, thereby recovering power via a generator and cooling the fluid further. A turbine bypass valve (E) operates in parallel to the turbine, allowing adjustment of the loop resistance to control the compressor operating point. After expansion, the cold fluid absorbs heat from a heat source or provides cooling in the low-temperature heat exchanger (F). Finally, the fluid passes through the recuperator (G) to be preheated before re-entering the compression system (A).

As investigated by Oehler, Gollasch, et al. (2021), the heat pump's fluid inventory (total working fluid mass) can be adjusted using a pair of valves. An inlet valve (H1) injects air from a high-pressure buffer tank to the low-pressure section to increase the fluid inventory, while a relief valve (H2) vents air from the high-pressure section to the ambient environment to decrease it. This fluid inventory control allows modulation of the heat pump's thermal output while maintaining the aerodynamic operating conditions of the turbomachinery. The demonstrated range for this control spans compressor inlet pressures (p_0) from 0.8 to 1.4 bar.

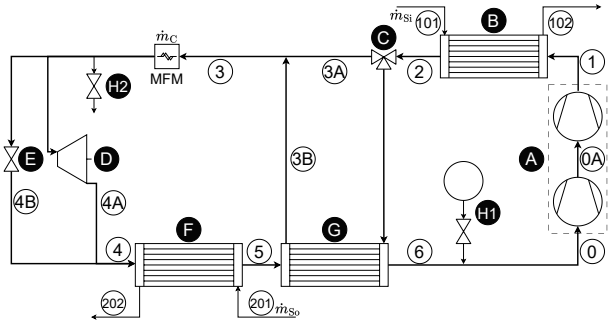


Figure 1. Flow schematic of the CoBra heat pump. (A) Two-stage compression system, (B) high-temperature heat exchanger, (C) three-way valve, (D) turbine, (E) turbine bypass valve, (F) low-temperature heat exchanger, (G) recuperator, (H1) buffer tank and air inlet valve, (H2) relief valve, MFM: Coriolis mass flow meter. Adapted from Tran and Stathopoulos (2025).

2.2 Dynamic model

A dynamic thermo-fluid model of CoBra heat pump was implemented in Modelica using the Dymola environment, as presented in previous work (Tran and Stathopoulos 2025), combining and extending models from the Modelica Standard Library (MSL), Buildings library (Wetter et al. 2014) and ThermoPower library (Casella and Leva 2006) to model the relevant thermodynamic processes and system dynamics. The model diagram is shown in Figure 2.

The libraries were selected due to their shared use of interfaces from the MSL, defined in `Modelica.Fluid` and `Modelica.Media` for fluid flow and properties, `Modelica.Thermal` for heat transfer, and `Modelica.Mechanics` for rotational mechanics. The specific contributions of each library to the overall model are as follows:

ThermoPower: The turbomachinery models (compressor and turbine) were implemented using models adapted from the ThermoPower library. These models assume steady-state behaviour without inherent mass or thermal dynamics, an assumption justified by the fast response time of the turbomachinery relative to other system components. Their behavior is defined by performance maps that relate pressure ratio, corrected mass flow, corrected shaft speed, and isentropic efficiency. The turbomachinery models also feature an `Modelica.Mechanics` flange, which enables a connection to the rotational components.

Buildings: The Buildings library was used to model the plant's piping, valves and fans. The volume and thermal inertia of the extensive piping network were represented using `MixingVolume` with dynamic energy and mass balances. Frictional losses in each pipe segment were modeled by two `PressureDrop` models in series, one with a quadratic ($\Delta p \sim \dot{m}^2$) and another with a linear ($\Delta p \sim \dot{m}$) pressure loss correlation. The three-way and turbine bypass valves are represented by the provided valve

DOI 10.3384/ecp2181017

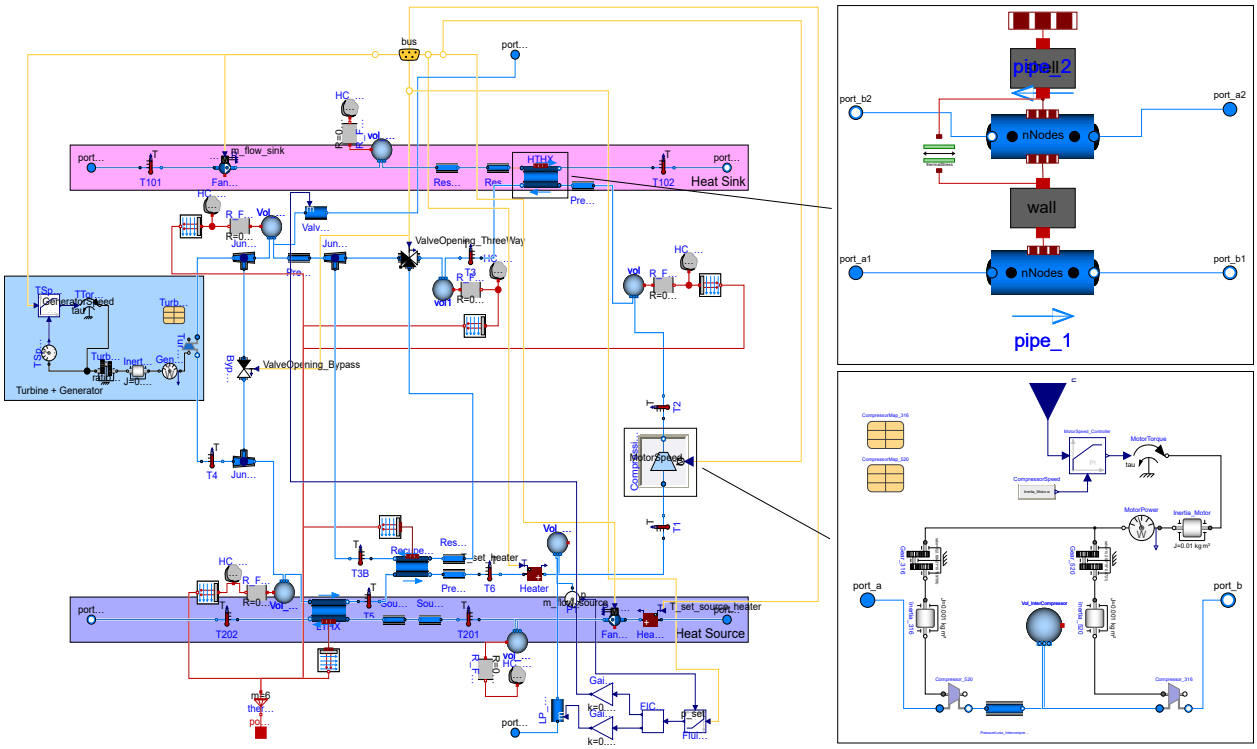


Figure 2. Diagram of the heat pump model in Dymola. Left: Overall heat pump. Top-right: Heat exchanger. Bottom-right: Compression system.

models with valve characteristics and sizings taken from manufacturer data. The fans providing air flow to the heat source and sink were represented by the library's flow-controlled mover models.

MSL: Beyond providing the interfaces, the MSL was employed to model mechanical components and heat exchangers. The rotational components, including the motor, generator, shafts and belt drives, were modeled using components from `Modelica.Mechanics`. The heat exchanger models were adapted from the `BasicHX` model and use a spatial discretization approach. In this method, the fluid flow paths and the solid metal structures (tubes and shell) are divided into multiple segments. For the purpose of this study, the baseline `BasicHX` model was extended to model several features of the real component: the heat capacity of the heat exchanger's outer shell, an additional heat port to model heat exchange with the ambient environment, and a model that calculates thermal stresses based on the temperature distribution along the tube and shell length.

2.3 Experimental validation

The methodology for calibrating the model against experimental data was presented in previous work (Tran and Stathopoulos 2025). Adjusting the compressor performance maps was a main focus, employing a transformation method aimed at minimizing prediction error while preserving the map's original shape, ensuring smoothness, and allowing for plausible extrapolation into unmeasured regions. Other model parameters, mainly heat transfer, heat loss and pressure loss coefficients, were tuned using numerical optimization to minimize deviations between simulation results and measurements.

For this study, a recalibration of the model was performed. This was necessary because new measurement data became available, recorded from a test setup that includes fully recuperated heat pump operation, a condition not covered in the previous work. An example test run from the new dataset is shown in Figure 3. In this test, following an initial ramp-up to nominal motor speed, a PI controller was applied to maintain the supply temperature (T_{102}) at 200 °C by adjusting compressor speed. Subsequently, the compressor inlet pressure (p_0) was varied stepwise (1.1 bar down to 0.8 bar in hourly intervals) using fluid inventory control. The test aimed to find the optimal operating point for the target supply temperature, indicated by the highest measured COP. The highest steady-state COP was measured at a pressure of 0.8 bar, which is the lower bound of investigated pressures. This suggests that the optimum might be at even lower pressures.

The predictions from the recalibrated model are compared to the measurement data shown in Figure 3. The resulting root mean squared errors (RMSE) and mean-normalized root mean squared errors (NRMSE) for the primary process variables are presented in Table 1. For the measured temperatures, pressures, and mass flow rate, the NRMSE values are in the range of 0.05 % to 1.95 %.

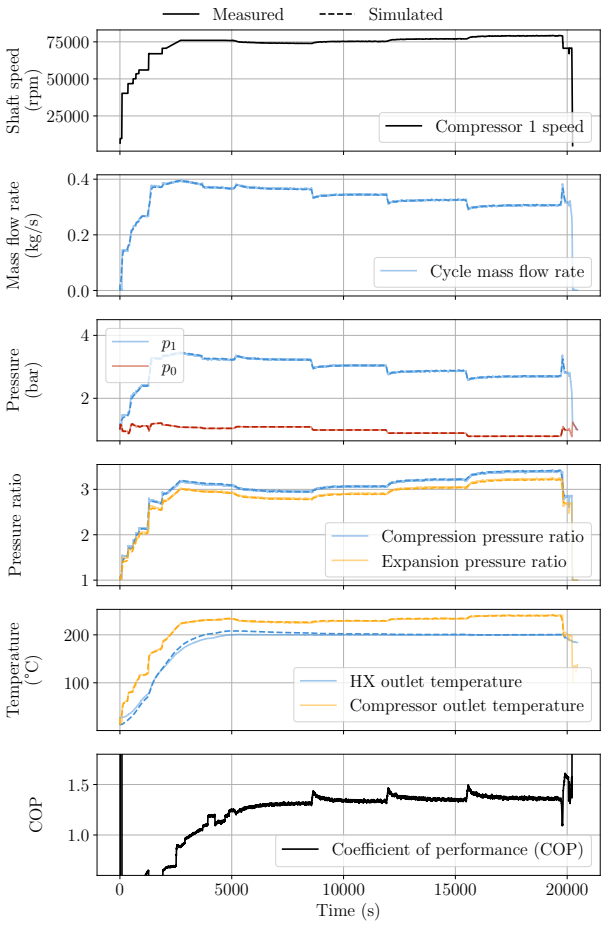


Figure 3. Comparison of measured (solid) and simulated (dashed) time series. The plot shows the alignment of experimental data and simulation results over time.

3 Heat Pump Control

The heat pump's target supply temperature (T_{102}), the return temperature from the process (T_{101}) and the heat sink mass flow rate ($\dot{m}_{\mathrm{Si}}$) are typically defined by process requirements. Consequently, a heat pump's control system is designed to track the set-point supply temperature while maintaining safe operating conditions. The CoBra heat pump features three main control variables that affect the supply temperature:

1. The *motor speed* (n_M) is the main manipulated variable for controlling the heat pump's supply temperature because it directly governs the compressor pressure ratio, mass flow rate and temperature lift.

2. The *three-way valve opening* (θ_{3W}) controls the amount of heat that is internally recovered by the recuperator. A value of $\theta_{3W} = 0$ represents full recuperation, whereas $\theta_{3W} = 1$ corresponds to fully non-recuperated operation.

3. Manipulation of the fluid inventory, indicated by the *compressor inlet pressure* (p_0), directly scales with the mass flow rate in the heat pump cycle, thereby providing control over its overall thermal output capacity.

Table 1. Root mean squared error (RMSE) and mean-normalized root mean squared error (NRMSE) of the main process variables.

Measurement signal		RMSE	NRMSE
Compressor outlet	T_1	1.64 K	0.33 %
HTHX outlet	T_2	3.89 K	1.19 %
Recuperator outlet (HP)	T_{3B}	0.72 K	0.25 %
Turbine inlet	T_3	0.56 K	0.19 %
Turbine outlet	T_{4A}	0.70 K	0.29 %
LTHX outlet	T_5	0.39 K	0.14 %
Recuperator outlet (LP)	T_6	1.73 K	0.56 %
HTHX inlet (sink)	T_{101}	0.15 K	0.05 %
HTHX outlet (sink)	T_{102}	4.82 K	1.06 %
LTHX inlet (source)	T_{201}	0.31 K	0.11 %
LTHX outlet (source)	T_{202}	1.23 K	0.49 %
Compressor inlet	p_0	1047 Pa	1.07 %
Compressor outlet	p_1	2337 Pa	0.79 %
Turbine inlet	p_3	2485 Pa	0.88 %
Mass flow rate	$\dot{m}$	0.0065 kg/s	1.95 %

During operation of the CoBra heat pump, two main safety constraints require monitoring. *Compressor surge* can occur when a compressor operates at low mass flow rates, potentially causing flow destabilization and detachment, which may lead to structural damage. However, the radial compressors currently installed in the heat pump feature broad performance maps, resulting in high inherent flow stability. Therefore, compressor surge is not considered a limiting operational constraint in this work. *Thermal stresses* arise in heat exchangers due to differences in thermal expansion of the tubes and the shell. Analyses of the HX model show that transient stresses are strongly correlated with the rate of heat accumulation, while steady-state stresses are correlated with the logarithmic mean temperature difference in the HX. Consequently, the rate of heat accumulation $\dot{Q}_{\mathrm{acc}}$ is used as a measurable proxy for these stresses. It is determined from the energy balance across the heat exchanger (HX) using measured mass flow rates and inlet/outlet temperatures, as defined in Equation 1:

$$\dot{Q}_{\mathrm{acc}} = (\dot{m}c_p)_{\mathrm{C}} \cdot (T_1 - T_2) - (\dot{m}c_p)_{\mathrm{Si}} \cdot (T_{102} - T_{101}) \quad (1)$$

A cascaded control strategy, shown in Figure 4, is applied to the supply temperature (T_{102}) while limiting the rate of heat accumulation ($\dot{Q}_{\mathrm{acc}}$). The outer loop controls T_{102} and sets the target accumulation rate ($\dot{Q}^{\mathrm{set}}_{\mathrm{acc}}$) for the inner loop, which adjusts the motor speed (n_M) accordingly. Limiting the outer loop's output combined with appropriate tuning of both controllers ensures that the thermal stress safety constraint is met.

The following three temperature control concepts are investigated in this work. Figure 5 shows schematics of the control concepts.

DOI
10.3384/ecp2181017

Concept 1: PI controllers This control approach utilizes proportional-integral (PI) controllers. Temperature control is achieved using the cascaded controller (`PI_T`) illustrated in Figure 4. In parallel, a separate PI controller (`PI_F`) adjusts p_0 to maintain a cycle-to-sink mass flow ratio of 1 ($\dot{m}_C/\dot{m}_{\mathrm{Si}} = 1$), as this ratio is typically near-optimal for minimizing heat transfer exergy losses. Throughout the operation, the three-way valve position is fixed at full recuperation ($\theta_{3W} = 0$), as heat recuperation is beneficial when the difference between T_2 and T_5 is sufficiently large to outweigh the additional pressure losses.

Concept 2: RL policy controller In this approach, a policy (usually a feed-forward neural network) trained through RL adjusts n_M, θ_{3W} and p_0 depending on the observed heat pump state and set-point value provided to the policy. Prior to deployment, the RL policy is trained to perform control actions that maximize an accumulated reward signal which incorporates the control task including the safety constraints.

Concept 3: PI and RL This approach combines traditional control and RL by using the cascaded controller for temperature control and a RL policy that learns to optimize the other control actions.

4 Reinforcement Learning

The objective of RL is for the agent to learn a policy π that maximizes the cumulative sum of rewards expected over time.

As described by Sutton and Barto (2018) and Achiam (2018), the standard RL setup involves an agent and an environment that interact over a series of time steps. At each step t, the agent receives an observation o_t from the environment. Based on this observation, which may be a subset of the full state s_t, the agent selects an action a_t according to its current policy π.

$$a_t = \pi(o_t) \tag{2}$$

In response to the action, the environment transitions to a new state s_{t+1} and provides the agent with a numerical reward r_t. This reward indicates the immediate value associated with taking action a_t in state s_t, as defined by a reward function R:

$$r_t = R(s_t, a_t) \tag{3}$$

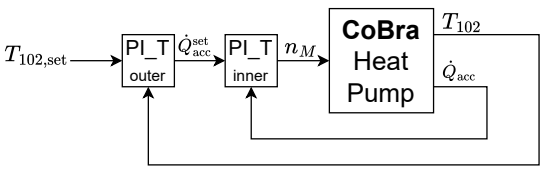
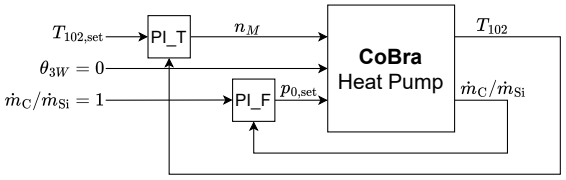
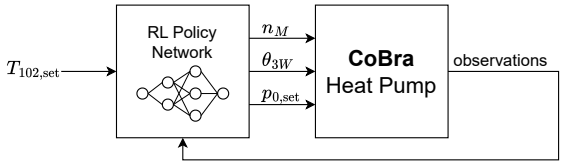


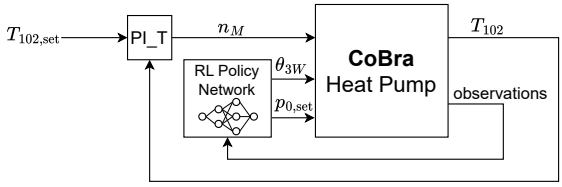
Figure 4. Cascaded controller for temperature control with limits for heat accumulation.



(a) Concept 1: PI controllers



(b) Concept 2: RL policy controller



(c) Concept 3: PI and RL

Figure 5. Control concepts for controlling the supply temperature (T_{102}) using the motor speed (n_M), three-way valve opening (θ_{3W}) and compressor inlet pressure (p_0).

4.1 Training setup

The RL training methodology in this work, depicted in Figure 6, employs established tools and standards for interfacing the heat pump model and the RL algorithm. The model is exported as a Co-Simulation Functional Mock-up Unit (FMU). A custom, Gymnasium-compliant (Towers et al. 2024) `Environment` interfaces with the FMU, providing motor speed, three-way valve opening, and compressor inlet pressure as control inputs.

The `Environment` uses the FMPy library to interface with the FMU. Following the modular design proposed by Tassa et al. (2018), problem-specific logic is delegated to a `Task` module, which calculates the reward based on the state and objective and adjusts simulation parameters and boundary conditions. This abstraction allows tasks to be interchanged - for instance, to switch between randomized and deterministic scenarios. Each episode begins from a "cold start" state, meaning all initial pressures and temperatures within the heat pump are set to ambient conditions.

Training is performed on episodes with a fixed length of 700 steps using a step size of 30 s, resulting in 21 000 s of simulated time per episode. This work utilizes the Stable-Baselines3 library (Raffin et al. 2021) which provides implementations of RL algorithms and offers parallelization and monitoring tools.

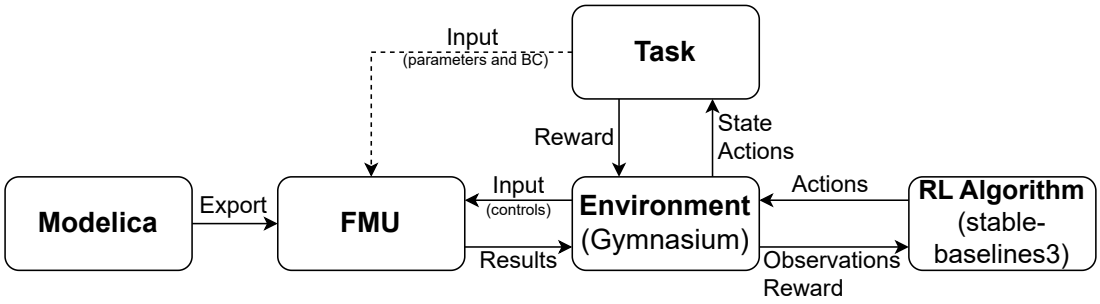


Figure 6. Task-supplied reinforcement learning using FMUs.

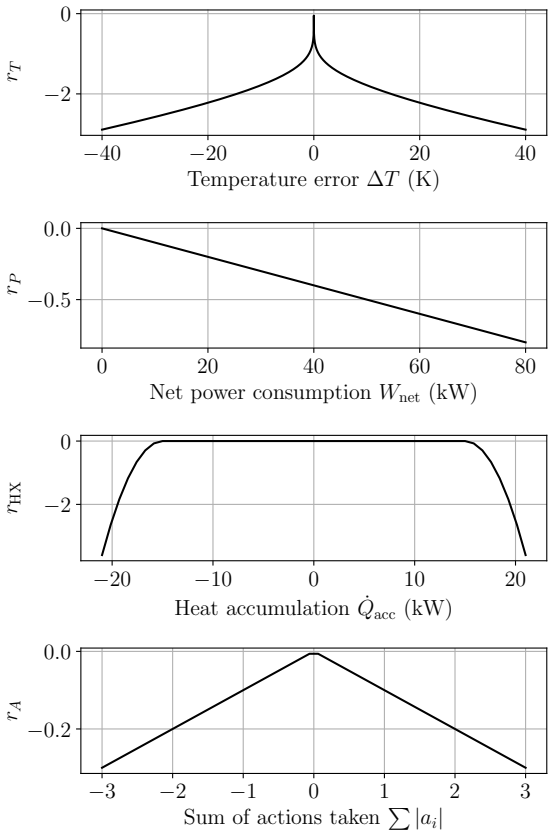


Figure 7. Components of the reward function.

4.2 Action and observation spaces

The action and observation spaces used for training are summarized in Table 2. As introduced in section 3, the action space consists of θ_{3W} and p_0 for both control concepts, with n_M additionally included for concept 2. Instead of outputting absolute values, the controller determines the desired *change* for each control input (delta action). The environment then adds this delta to the current value. This approach inherently constrains the slew rates of the actuators, while the simulation model enforces separate limits on the final values to keep them within their allowable operational range. For this study, the lower bound for compressor inlet pressure (p_0) was lowered from 0.8 bar to 0.55 bar. This adjustment was made because experimental results suggest the optimal p_0 value might be below the original 0.8 bar limit. The observation space includes the actual actuator signals, the rate of heat accumulation, pressure ratio, and various temperature and flow signals from the plant. All signals from the observation space are measured in the physical plant.

Table 2. Action and observation spaces.

Action space	
Change of motor speed	Δn_M
Change of three-way valve opening	$\Delta\theta_{3W}$
Change of compressor inlet pressure	Δp_0
Observation space	
Temperature control error	ΔT
Rate of heat accumulation	$\dot{Q}_{\text{acc}}$
Motor speed	n_M
Three-way valve opening	θ_{3W}
Compressor inlet pressure	p_0
Cycle mass flow rate	$\dot{m}_C$
Sink mass flow rate	$\dot{m}_{\text{Si}}$
Sink inlet temperature	T_{101}
Sink outlet temperature	T_{102}
Compressor outlet temperature	T_1
HTHX outlet temperature	T_2
Compressor pressure ratio	Π_C
Coefficient of performance	COP

4.3 Reward function

A major challenge in RL is the reward function design: maximizing the reward must yield the desired control behavior, yet the function should also guide the agent's learning. The reward function in Equation 4 used in this work is a sum of four reward components. Each component is defined as a penalty term (a negative reward) representing a training goal, and the agent maximizes the total

DOI
10.3384/ecp2181017

reward by learning to minimize these penalties.

$$R = r_T + r_P + r_{\text{HX}} + r_A \tag{4}$$

$$r_T = b_T \cdot \left(|\Delta T|^{0.2} + 0.02 \cdot |\Delta T|\right) \tag{5}$$

$$r_P = b_P \cdot W_{\text{net}} \tag{6}$$

$$r_{\text{HX}} = \begin{cases} b_{\text{HX}} \cdot \left(|\dot{Q}_{\text{acc}}| - \dot{Q}_{\text{acc}}^{\text{lim}}\right)^2 & \text{if } |\dot{Q}_{\text{acc}}| > \dot{Q}_{\text{acc}}^{\text{lim}} \\ 0 & \text{otherwise} \end{cases} \tag{7}$$

$$r_A = b_a \sum_i |a_i| \tag{8}$$

For the temperature tracking goal r_T, the penalty is calculated based on the control error ΔT. It includes terms proportional to $|\Delta T|^{0.2}$ and $|\Delta T|$. This structure leads to the behaviour shown in Figure 7. The linear term ($|\Delta T|$) provides a consistent gradient signal while the power term ($|\Delta T|^{0.2}$) creates a steep reward gradient near the setpoint. This discourages the agent from trading off precise temperature tracking for the conflicting goal of energy minimization.

The r_P component directly penalizes net power consumption W_{net}, thereby incentivizing energy-efficient operation.

r_{HX} penalizes the rate of heat accumulation ($\dot{Q}_{\text{acc}}$) if it exceeds a soft limit, $\dot{Q}_{\text{acc}}^{\text{lim}}$. The penalty increases quadratically with the level of exceedance. This design makes minor and brief overshoots more acceptable than large and sustained violations.

r_A penalizes the magnitude of the control actions taken, discouraging excessive actuation and promoting smoother control.

The overall behavior is shaped by the weighting coefficients (b_T, b_P, b_{HX}, b_a) within the reward components. These coefficients are carefully selected to prioritize certain objectives. For example, maintaining the target temperature (r_T) and ensuring heat exchanger safety (r_{HX}) were weighted more heavily than optimizing for energy efficiency (r_P) or minimizing control action (r_A). In this work, good learning results were found for coefficient values of $b_T = -1$, $b_P = -1 \times 10^{-5}$, $b_{\text{HX}} = -1 \times 10^{-7}$ and $b_a = -0.1$.

4.4 Domain randomization

Domain randomization is a method proposed by Tobin et al. (2017), addressing the Sim2Real challenge when deploying the agent on the real system. The idea of domain randomization is to expose the RL algorithm to variance (e.g. system dynamics, set-points, boundary conditions) during training in order to train a model that generalizes and works across all variants. In this work, randomization is implemented in two ways:

Parameter variation: At the beginning of each training episode, the model parameters (including heat transfer characteristics, turbomachinery map scalers, and actuator rise times) are sampled from uniform distributions to introduce variability. The bounds for this sampling are set symmetrically around each parameter's nominal value, defining a range that is likely to contain the true physical value. The width of this range is selected according to the estimated uncertainty for each parameter. For example, well-calibrated parameters like turbomachinery map scalers are varied over a narrow range ($\pm 2\,\%$), while more uncertain parameters such as heat transfer scalers and actuator delays are given a wider range (up to $\pm 40\,\%$).

Set-point variation: Within each episode, the temperature set-point $T_{102,\text{set}}$ is varied. It follows diverse signal types including steps, ramps, constant values, harmonics, and random walks within the heat pump's temperature operating range.

4.5 Training with Soft Actor-Critic

Training was conducted using Soft Actor-Critic (SAC) developed by Haarnoja et al. (2018). SAC is an off-policy actor-critic algorithm in which a policy network, the *actor*, is optimized using feedback from a value function network, the *critic*. It is based on the maximum entropy RL framework, where the policy is trained to maximize and balance entropy (exploration) and expected returns (exploitation) by automatically adjusting the entropy coefficient. This algorithm was selected due to its sample efficiency and demonstrated suitability for continuous control tasks.

Hyperparameters for training were determined by hyperparameter optimization using RL Baselines3 Zoo (Raffin 2020). In addition to the tuning results, a linear learning rate schedule was added, starting higher for rapid initial progress and decreasing over time to allow for finer policy adjustments. The hyperparameters used to produce the presented results are summarized in Table 3.

Table 3. Hyperparameters used for training.

Parameter	*Value*
Learning rate	5×10^{-5} to 5×10^{-6}
Parallel environments	8
Training frequency	1 per step
Gradient steps	8
Batch size	512
Replay buffer size	1×10^6
Discount rate	0.99
Entropy coefficient	Auto
Actor network	2 layers, each 128 neurons
Critic network	2 layers, each 256 neurons

To accelerate training, the training process was parallelized across 8 environments using the `SubProcEnv` implementation of Stable-Baselines3. To assess the effect of random initialization, all training was conducted independently on multiple agents, each for 4 million environment steps. A single training run required approximately 36 hours on a scientific computing workstation, and agents were saved periodically using the library's callback mechanism.

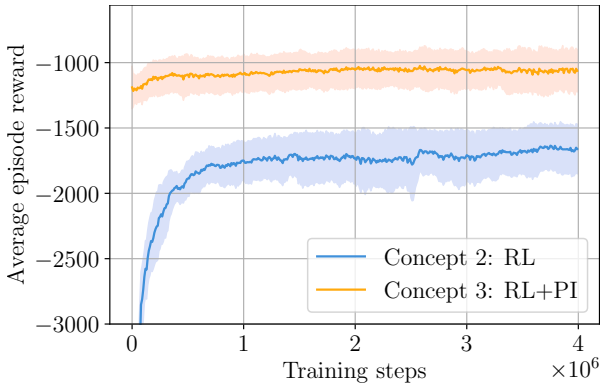


Figure 8. Average episode reward during training. Solid line: Average reward across all agents. Shaded area: Standard deviation of reward.

5 Results

This section evaluates the proposed control concepts, beginning with an analysis of the RL agents' training progress. All three concepts are then evaluated on a nominal baseline model, where the RL+PI agent demonstrates a novel and efficient control strategy developed during training. Finally, the RL agents' robustness and generalization capabilities are tested against a range of randomized model variations.

5.1 Training progress

Figure 8 shows the training performance, plotting the mean episode reward (solid line) and standard deviation (shaded area) across all runs.

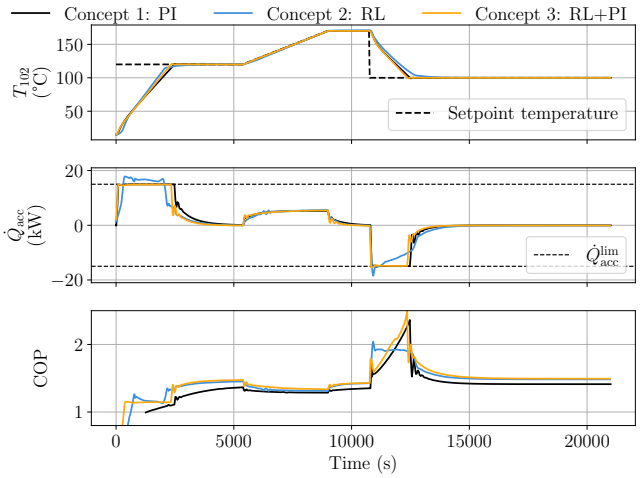
The learning curves reveal different training dynamics. Concept 2 shows a standard learning progression, with rewards rising steeply as the agent learns the control tasks from scratch. In contrast, concept 3 begins with a high initial reward that improves only marginally. This is because its embedded PI controller already handles the primary temperature control objective. The RL agent's role is thus limited to optimizing secondary, lower-weighted reward terms for efficiency and control actions.

The fluctuations observed in both curves likely stem from the inherent stochasticity of the training process, amplified by domain randomization.

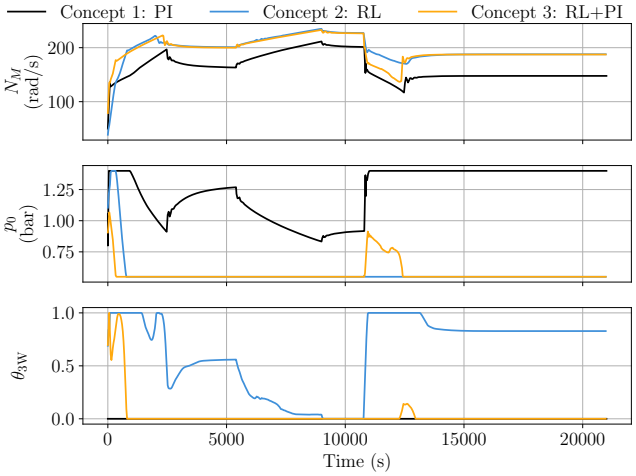
5.2 Control performance on baseline model

To directly compare the control performance of the three concepts, the best-performing agent for each was evaluated in a deterministic, non-randomized environment. The agents were tasked with tracking a temperature set-point trajectory not seen during training, as shown in Figure 9a. This trajectory includes a step increase to 120 °C, a linear ramp to 170 °C, a hold at that temperature, and a final step decrease to 100 °C.

Figure 9 shows the control and process variables for all control concepts. While concepts 1 and 3 rely on embedded PI controllers for temperature tracking and heat accumulation, concept 2's RL policy learned to perform



(a) Process variables



(b) Control variables

Figure 9. Comparison of the three control concepts for temperature tracking with a varying set-point signal on the baseline model, showing (a) process variables and (b) control variables.

these tasks, showing relatively accurate temperature control. Regarding the thermal stress constraint, concept 3 strictly maintains the soft limit for heat accumulation ($\dot{Q}_{acc}^{lim}$), whereas concept 2 exceeds this limit by a slight but acceptable margin.

Notably, concept 3 achieves the highest efficiency, maintaining a higher COP for nearly the entire duration of the maneuver. This performance is the result of a novel policy discovered by the RL agent. Similar to the baseline in concept 1, this policy keeps the three-way valve opening (θ_{3W}) fixed at 0 (fully recuperated operation). However, it manipulates the compressor inlet pressure (p_0), minimizing it during steady-state operation and increasing it during transients. This pressure manipulation strategy is noteworthy for two reasons:

1. *Steady-state operation:* The decision to lower p_0 is initially counterintuitive. A lower inlet pressure is creates imbalanced mass flow ratios in the heat exchanger, which increases entropy generation of heat

DOI
10.3384/ecp2181017

transfer. However, the observed improvement in COP indicates that the efficiency gains from reduced overall pressure losses successfully outweigh the increased exergy losses from heat transfer.

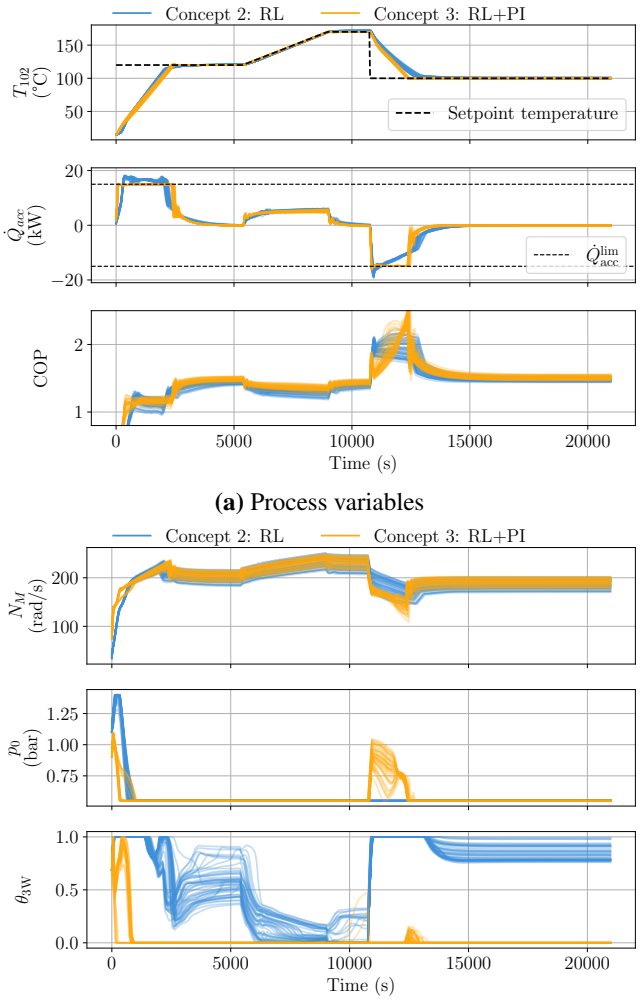
2. *Transient operation:* The strategy of increasing p_0 when ΔT is high (e.g. after a set-point change) is a particularly surprising discovery. Further analysis reveals that this action allows the system to converge to the new temperature set-point faster without exceeding the soft limit on heat accumulation.

Although further performance gains are anticipated with longer training and hyperparameter optimization, these findings confirm that RL can be successfully integrated with traditional PI controllers to discover novel and more efficient control policies.

5.3 Control performance on model variations

To assess the robustness of the learned policies against uncertainties of the system properties and dynamics, the best-performing agent from each RL control concept was evaluated on 50 model variations. These variations were sampled from the parameter range used for domain randomization.

The results from all 100 evaluation runs (50 for each RL concept) are illustrated in Figure 10. The figure shows that both RL concepts maintain temperature tracking and thermal stress limitation across the entire set of variations. Furthermore, the general control strategies employed by the agents remain consistent with those observed under nominal conditions, as described in subsection 5.2. This suggests that the RL agents are capable of generalizing to varied system properties and can maintain control performance in the presence of model uncertainty.



(a) Process variables

(b) Control variables

Figure 10. Comparison of the RL control concepts for temperature tracking with a varying set-point signal on randomized model variations, showing (a) process variables and (b) control variables.

6 Conclusion

This study investigated RL for controlling a high-temperature Brayton cycle heat pump, using an experimentally validated model. Different control concepts were compared: traditional PI controllers (Concept 1), a pure RL policy (Concept 2), and a hybrid PI+RL approach (Concept 3). Training was performed with the Soft Actor-Critic algorithm, and domain randomization was employed to improve controller robustness against parameter uncertainties.

The results indicate that while the pure RL agent learned the fundamental control tasks, the hybrid PI+RL approach (Concept 3) achieved higher operational efficiency. This improved performance was attributed to the discovery of a novel strategy for manipulating the compressor inlet pressure (p_0). The learned policy minimizes p_0 during steady-state operation to maximize efficiency, while temporarily increasing it during transients. This transient pressure increase, applied when the set-point temperature was changed, was a particularly noteworthy finding. It enables the system to converge to new set-points faster without exceeding thermal stress constraints. The controllers also demonstrated robust operation across a range of model parameters.

While the simulation results are positive, they do not serve as a guarantee of successful real-world deployment. The transfer and validation of the controller on the physical system are therefore planned for future work.

In summary, the findings suggest that RL, particularly when paired with traditional PI(D) controllers, is a viable method for controlling Brayton heat pumps and has the potential to increase their efficiency.

References

Achiam, Joshua (2018). *Spinning Up in Deep Reinforcement Learning*. URL: https://spinningup.openai.com/en/latest/index.html (visited on 2025-06-27).

Åkesson, Johan (2008). "Optimica—An Extension of Modelica Supporting Dynamic Optimization". English. In: *Proceed-*

ings of the 6th International Modelica Conference. The 6th International Modelica Conference (March 3–4, 2008).

Andersson, Joel A. E. et al. (2019). "CasADi: a software framework for nonlinear optimization and optimal control". In: *Mathematical Programming Computation* 11.1. PII: 139, pp. 1–36. ISSN: 1867-2949. DOI: 10.1007/s12532-018-0139-4.

Casella, Francesco and Alberto Leva (2006). "Modelling of thermo-hydraulic power generation processes using Modelica". In: *Mathematical and Computer Modelling of Dynamical Systems* 12.1, pp. 19–33. ISSN: 1387-3954. DOI: 10.1080/13873950500071082.

Di Cao et al. (2020). "Reinforcement Learning and Its Applications in Modern Power and Energy Systems: A Review". In: *Journal of Modern Power Systems and Clean Energy* 8.6, pp. 1029–1042. ISSN: 2196-5625. DOI: 10.35833/MPCE.2020.000552.

Haarnoja, Tuomas et al. (2018). "Soft Actor-Critic: Off-Policy Maximum Entropy Deep Reinforcement Learning with a Stochastic Actor". In: *Proceedings of the 35th International Conference on Machine Learning*. Ed. by Jennifer Dy and Andreas Krause. Vol. 80. Proceedings of Machine Learning Research. 10–15 Jul. PMLR, pp. 1861–1870. URL: https://proceedings.mlr.press/v80/haarnoja18b.html.

Heat Pump Centre (2023). *IEA HPT Annex 58. High-Temperature Heat Pumps*. URL: https://heatpumpingtechnologies.org/annex58/ (visited on 2024-04-25).

IEA (2024). *World Energy Outlook 2024*. Ed. by IEA. Paris. URL: https://www.iea.org/reports/world-energy-outlook-2024 (visited on 2024-10-31).

Lukianykhin, Oleh and Tetiana Bogodorova (2019). "ModelicaGym: Applying Reinforcement Learning to Modelica Models". In: *Proceedings of the 9th International Workshop on Equation-based Object-oriented Modeling Languages and Tools*. EOOLT '19: 9th International Workshop on Equation-Based Object-Oriented Modeling Languages and Tools (Berlin Germany, November 5, 2019). Ed. by Christoph Nytsch-Geusen and Olaf Enge-Rosenblatt. New York, NY, USA: ACM, pp. 27–36. ISBN: 9781450377133. DOI: 10.1145/3365984.3365985.

Magnusson, Fredrik and Johan Åkesson (2015). "Dynamic Optimization in JModelica.org". In: *Processes* 3.2. PII: pr3020471, pp. 471–496. DOI: 10.3390/pr3020471.

Modelon, Inc. (2024). *OPTIMICA Compiler Toolkit*. URL: https://help.modelon.com/latest/reference/oct/ (visited on 2025-07-30).

Oehler, Johannes, Jens Gollasch, et al. (2021). "Part Load Capability of a High Temperature Heat Pump with Reversed Brayton Cycle". In: *13th IEA Heat Pump Conference 2021 (HPC2020) Conference Proceedings*. 13th IEA Heat Pump Conference (Jeju, Korea, April 26–29, 2021). Ed. by International Energy Agency.

Oehler, Johannes, A. Phong Tran, and Panagiotis Stathopoulos (2022). "Simulation of a Safe Start-Up Maneuver for a Brayton Heat Pump". In: *Volume 4: Cycle Innovations; Cycle Innovations: Energy Storage*. ASME Turbo Expo 2022: Turbomachinery Technical Conference and Exposition (Rotterdam, Netherlands, June 13–17, 2022). American Society of Mechanical Engineers. ISBN: 978-0-7918-8601-4. DOI: 10.1115/GT2022-79399.

Raffin, Antonin (2020). *RL Baselines3 Zoo*. URL: https://github.com/DLR-RM/rl-baselines3-zoo (visited on 2025-06-27).

Raffin, Antonin et al. (2021). "Stable-Baselines3: Reliable Reinforcement Learning Implementations". In: *Journal of Machine Learning Research* 22.268, pp. 1–8. URL: http://jmlr.org/papers/v22/20-1364.html.

Silver, David et al. (2018). "A general reinforcement learning algorithm that masters chess, shogi, and Go through self-play". eng. In: *Science (New York, N.Y.)* 362.6419. Journal Article Research Support, Non-U.S. Gov't, pp. 1140–1144. DOI: 10.1126/science.aar6404. eprint: 30523106.

Sutton, Richard S. and Andrew Barto (2018). *Reinforcement learning. An introduction*. eng. Second edition. Adaptive computation and machine learning. Cambridge, Massachusetts and London, England: The MIT Press. 526 pp. ISBN: 978-0262039246.

Tang, Chen et al. (2025). "Deep Reinforcement Learning for Robotics: A Survey of Real-World Successes". In: *Annual Review of Control, Robotics, and Autonomous Systems* 8.1, pp. 153–188. DOI: 10.1146/annurev-control-030323-022510.

Tassa, Yuval et al. (2018). *DeepMind Control Suite*. DOI: 10.48550/arXiv.1801.00690.

Tobin, Josh et al. (2017). "Domain randomization for transferring deep neural networks from simulation to the real world". In: *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (Vancouver, BC, September 24–28, 2017). IEEE, pp. 23–30. ISBN: 978-1-5386-2682-5. DOI: 10.1109/IROS.2017.8202133.

Towers, Mark et al. (2024). *Gymnasium: A Standard Interface for Reinforcement Learning Environments*. DOI: 10.48550/arXiv.2407.17032.

Tran, A. Phong and Panagiotis Stathopoulos (2025). "Dynamic simulation and experimental validation of a high-temperature Brayton heat pump". In: *Applied Thermal Engineering* 274. PII: S1359431125011287, p. 126536. ISSN: 13594311. DOI: 10.1016/j.applthermaleng.2025.126536.

Wetter, Michael et al. (2014). "Modelica Buildings library". In: *Journal of Building Performance Simulation* 7.4, pp. 253–270. ISSN: 1940-1493. DOI: 10.1080/19401493.2013.765506.

Wrede, Konstantin et al. (2024). "FMUGym: An Interface for Reinforcement Learning-based Control of Functional Mock-up Units under Uncertainties". In: *31st International Workshop on Intelligent Computing in Engineering, EG-ICE 2024*, pp. 647–656. DOI: 10.35869/Proceedings_EGICE2024. URL: https://publica.fraunhofer.de/entities/publication/d31204fd-3480-469f-9c45-5c376d81a721/fullmeta.

Yücel, Fatma Cansu et al. (2025). "Design and Commissioning of the Brayton High-Temperature Heat Pump "Cobra"". In: DOI: 10.2139/ssrn.5290358.

Zhao, Wenshuai, Jorge Pena Queralta, and Tomi Westerlund (2020). "Sim-to-Real Transfer in Deep Reinforcement Learning for Robotics: a Survey". In: *2020 IEEE Symposium Series on Computational Intelligence (SSCI)*. 2020 IEEE Symposium Series on Computational Intelligence (SSCI) (Canberra, ACT, Australia, December 1–4, 2020). IEEE, pp. 737–744. ISBN: 978-1-7281-2547-3. DOI: 10.1109/SSCI47803.2020.9308468.

Zühlsdorf, B. et al. (2019). "Analysis of technologies and potentials for heat pump-based process heat supply above 150 °C". In: *Energy Conversion and Management: X* 2. PII: S2590174519300091, p. 100011. ISSN: 25901745. DOI: 10.1016/j.ecmx.2019.100011.