

Chenzi Huang, Stephan Seidel

Reinforcement Learning und Modellprädiktive Regelung für smarte Steuerungsfunktionen im Gebäudeenergiesystem

ARCHE Abschlussworkshop. Dresden. 27|09|22

AGENDA

- Motivation für einen Energiemanager
- Demonstrator und Modellierung
- Reinforcement Learning - RL
 - Funktion
 - Implementierung
 - Lernen & Gewinn
 - Lernoptimierung
 - Erweiterung
- Ausblick

Motivation

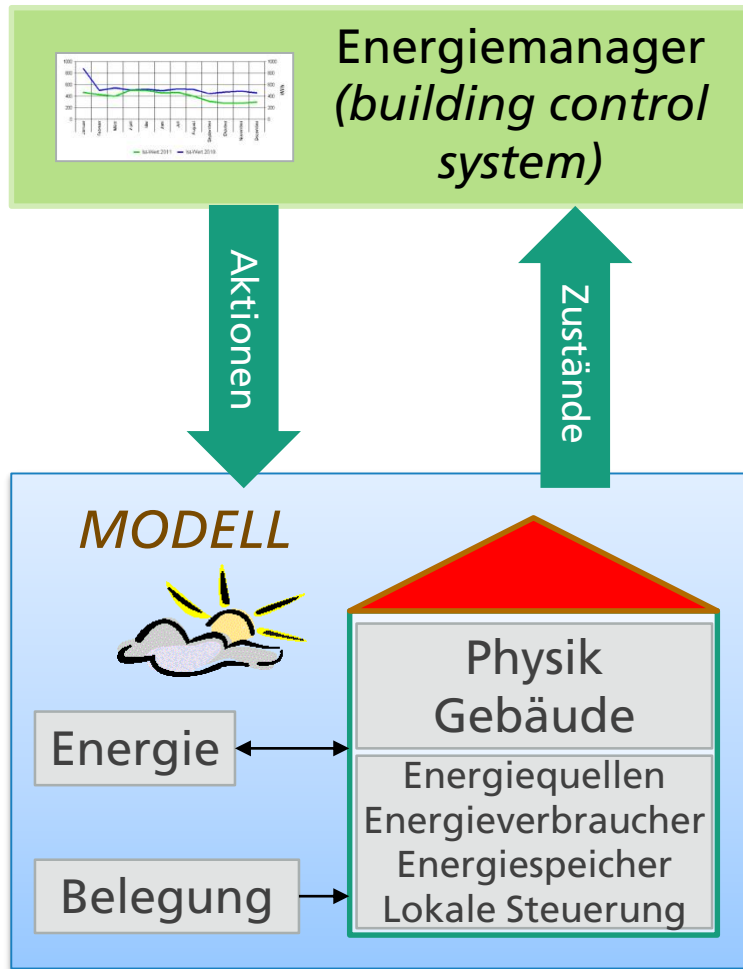


- Klimaneutraler Gebäudebestand im Jahr 2050 (Deutschland)
- Aktuelle Entwicklung in 2022 (Ukraine, Erdgas) beschleunigt Änderungsprozess
- Energiewende und damit verbundene Förderprogramme, z.B. "Energiewende Bauen"
- Säulen → Erneuerbare Energien und Energieeffizienz
- Energieeffizienz in Gebäuden
 - Energieverlust
 - Verbesserung der Gebäudedämmung (→ Limit erreicht)
 - Energieverbrauch
 - Verbrauchssenkung durch intelligente Steuerungssysteme

Gebäudeautomatisierungssysteme

- Standard-Lösungen sind wenig energieeffizient
- Orientierung auf Erfüllung der Versorgungsaufgabe
- Meist Inselösungen ohne überlagerte Steuerung zur Orchestrierung des Gesamtsystems
- Nur wenige moderne Verfahren im Einsatz
- Multivalente Energiesysteme erfordern jedoch eine überlagertes Steuerungssystem
- Moderne Verfahren aus Informatik und Mathematik bieten sich hierbei an:
 - Künstliche Intelligenz – Reinforcement Learning
 - Modellprädiktive Regelung
- Ziel:
 - Entwicklung eines Energieverteilmanagers → Vorrasschauende und möglichst effiziente Steuerung der unterlagerten Energiesystemkomponenten

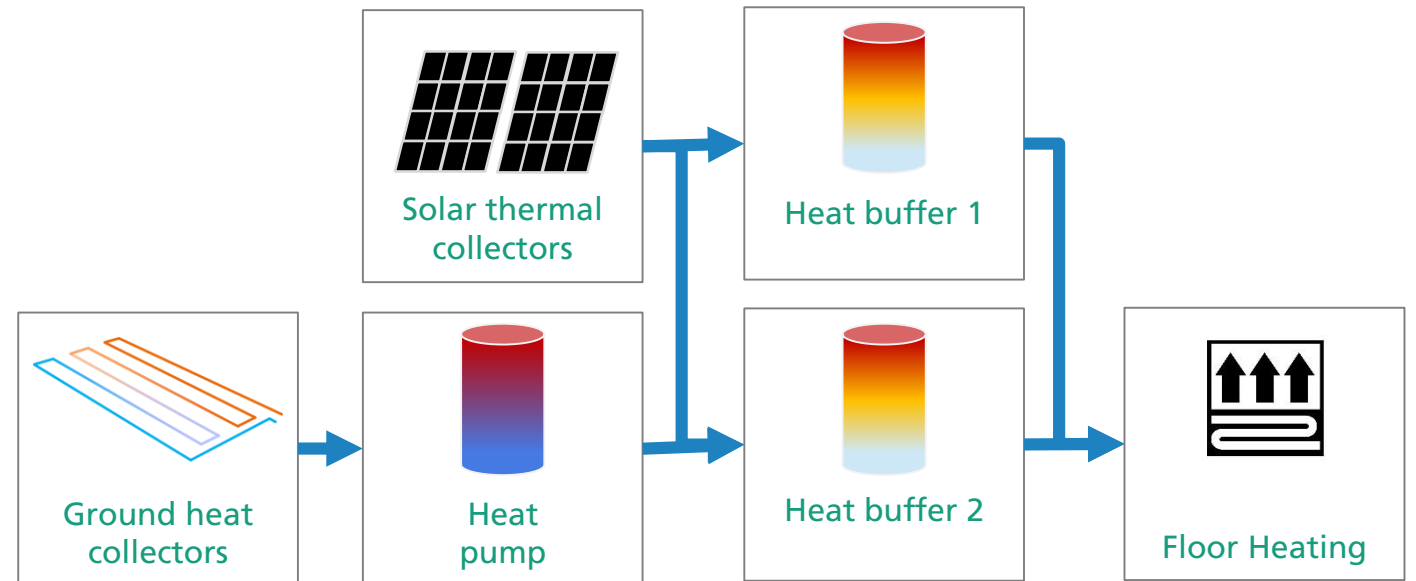
Energie(verteil-)manager - Motivation



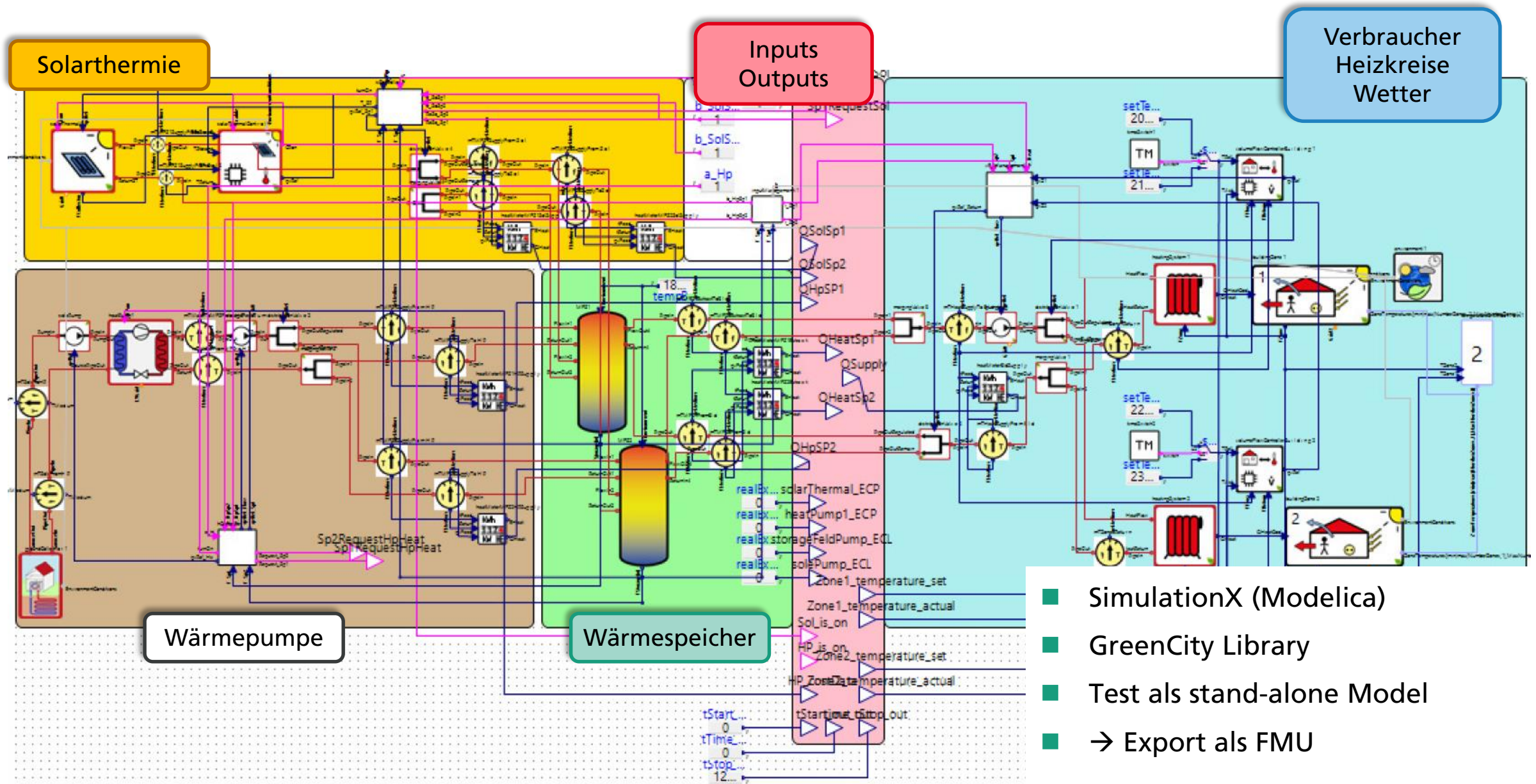
- Konzeption eines konventionellen Energieverteilmanagers
 - Regelbasierter Ansatz
 - klassische Steuerungstechnik / Automatengraph
 - Herausforderung: Integration von Prognosen zu Wetter, Bedarf, Störgrößen
- Konzeption eines smarten Energieverteilmanagers
 - Zwei Ansätze
 - **MPC** - Modellprädiktive Regelung
 - **RL** - Reinforcement Learning
- Beide Ansätze benötigen ein ausführbares Systemmodell
 - Modellierung des Gebäudes
 - Modellierung des Energiesystems-/komponenten
 - Modellierung der unterlagerten Steuerungen

Modellierung des Demonstrators ifm-Musterhaus

- Bivalentes Energiesystem
 - Wärmepumpe mit Flächenkollektor
 - Solarthermie
 - 2 Wärmespeicher
 - 2 Gebäudezonen mit Fußbodenheizung
 - Lokale Steuerungen
- Modellierung mit SimulationX in Modelica unter Verwendung der GreenCity Library
- Modellierung mit lokalen Steuerungen ohne Interaktion zwischen Solarthermie und Wärmepumpe



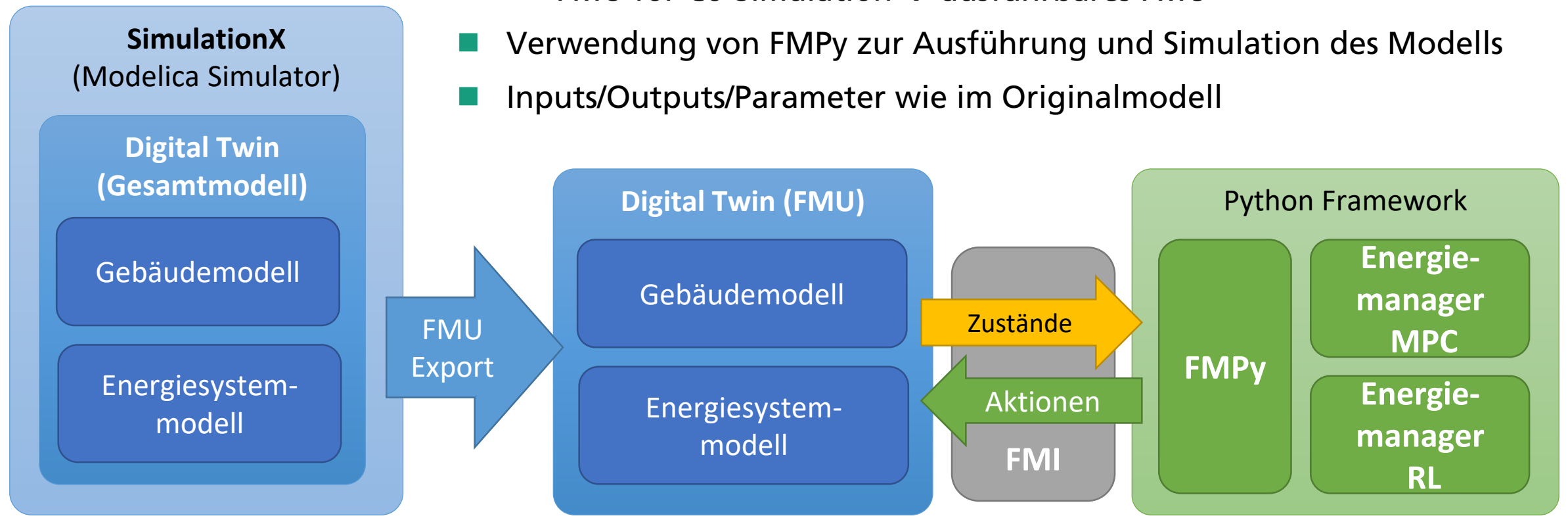
SimulationX Modell des Demonstrator-Gebäudes



- SimulationX (Modelica)
- GreenCity Library
- Test als stand-alone Model
- → Export als FMU

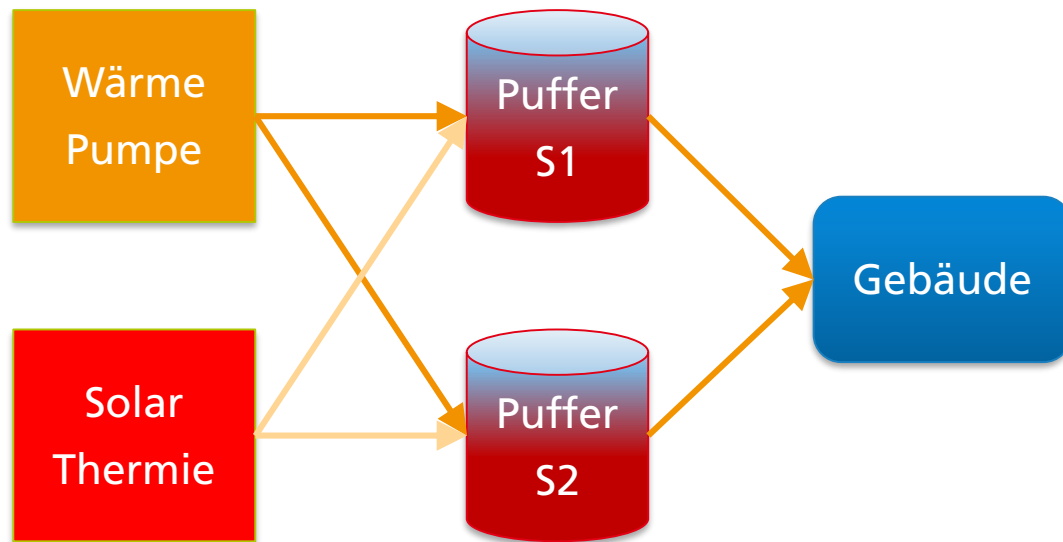
Integration des Modells in die Entwicklungsumgebung

- Export des Modells als extern ausführbare Functional Mock-Up Unit (FMU) aus SimulationX
 - FMU for Co-Simulation → ausführbares FMU
- Verwendung von FMPy zur Ausführung und Simulation des Modells
- Inputs/Outputs/Parameter wie im Originalmodell



Energieverteilmanager

- Aufgaben des Energieverteilmanagers
 - Optimierung des Energieflusses im System
 - Minimierung der Energiekosten
 - Beibehaltung des Komforts



- Ausgangssignale:
 - α_{HP} : Freigabe WP on/off
 - α_{Sol} : Freigabe Solarthermie
 - $\beta_{HpS1}, \beta_{HpS2}$: Vorwahl WP zu S1 o. S2
 - $\beta_{SolS1}, \beta_{SolS2}$: Vorwahl Solar zu S1 o. S2
 - γ_{S1}, γ_{S2} : S1 oder S2 zur Wärmevers.

→ 8 binäre Ausgänge

$$\mathbf{u} = (\alpha_{HP}, \beta_{HpS1}, \dots, \alpha_{Sol}, \dots, \gamma_{S1}, \gamma_{S2})^T,$$

$$\mathbf{u} \in U = \{0, 1\}^8$$

Arten des Maschinellen Lernens (Machine Learning)

Supervised Learning

- Beobachtung und Korrektur durch einen Mentor (External Supervision)
- Maschinelles Lernen mit Aufsicht und Regeln

Reinforcement Learning

- Selbsttätige Optimierung durch Spielen und Ausprobieren
- Bestärkendes Lernen durch Belohnungen

Unsupervised Learning

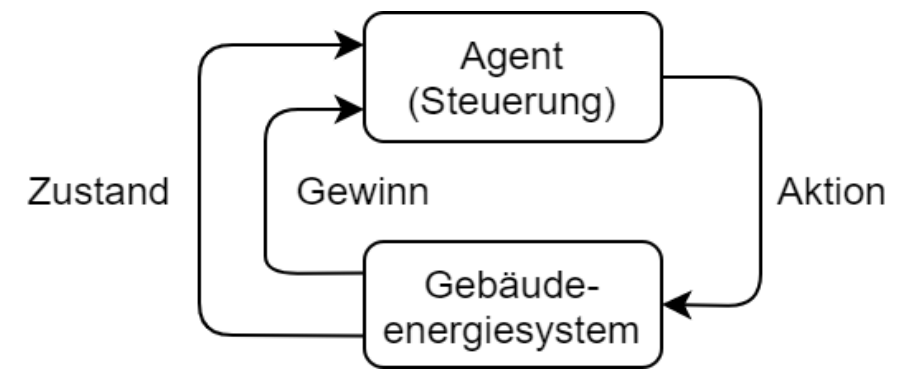
- Mustererkennung in Datenmengen (Erkennen und Auswerten)

Deep Learning

- Nutzung Neuronaler Netze in Analogie zum menschlichen Gehirn

Reinforcement Learning

- RL-Agent lernt was zu tun ist (Zuordnung von Aktionen zu Zuständen/Situationen)
- Agent erlernt, welche Aktion in einem Zustand bringt die größte Belohnung bzw. einen Fehler hervorruft
- Agent folgt der Trial & Error Methode, wobei er lernt und entscheidet
- Agent interagiert zyklisch mit der Umgebung (hier: Gebäudeenergiesystem)
- Zyklische Interaktion:
 - Agent entscheidet sich für eine Aktion
 - Umgebung reagiert auf Aktion mit einem neuen Zustand
 - Agent erkennt neuen Zustand und bestimmt den Gewinn
 - Agent entscheidet sich für eine Folgeaktion



Reinforcement Learning als Energiemanager

- Reinforcement Learning als Q-Learning implementiert
- Q-Tabelle enthält Zustände und Aktionen sowie den erwartbaren Gewinn
- Zellen der Q-Tabelle werden nach einer ausgeführten Aktion aktualisiert

- $$Q_t(s, a) = Q_{t-1}(s, a) + \alpha \left(R(s, a) + \gamma \max_{a'} Q(s', a') - Q_{t-1}(s, a) \right)$$

- Zu Beginn des Lernens: Alle Zellen = 0
- Auswahl der Aktion erfolgt nach ϵ -greedy Verfahren
 - ϵ Faktor bestimmt Auswahl einer zufällig Aktion oder aber der Aktion mit dem höchsten Gewinn
 - Somit Sicherstellung, dass Agent weiterhin lernt und nicht nur bekannte Aktionen wählt

Q_t	neuer Q-Wert
Q_{t-1}	alter Q-Wert
R	Gewinn
γ	Diskontfaktor
$\max Q$	maximaler zukünftig erwarteter Gewinn aller Aktionen im Folgezustand

RL Algorithmen

Q-Learning vs. SARSA

Q-Learning

Wiederholung (RL Lernprozess):

Beobachten von aktueller Belohnung R und Zustand $s' \in S$

$$Q^{new}(s, a) \leftarrow Q^{old}(s, a) + \alpha [R + \gamma \cdot \max_{a \in A} Q(s', a) - Q^{old}(s, a)]$$

$$s \leftarrow s'$$

Auswählen $a' \in A$ mittels ϵ -greedy Policy

$$a \leftarrow a'$$

SARSA

Wiederholung (RL Lernprozess):

Beobachten von aktueller Belohnung R und Zustand $s' \in S$

Auswählen $a' \in A$ mittels ϵ -greedy Policy

$$Q^{new}(s, a) \leftarrow Q^{old}(s, a) + \alpha [R + \gamma \cdot Q(s', a') - Q^{old}(s, a)]$$

$$s \leftarrow s'$$

$$a \leftarrow a'$$

Reinforcement Learning als Energiemanager

■ s – Zustände des Energiesystems

- $s = [Occupancy, Charge_{sp1}, Charge_{sp2}, Temperature_{outside}]^T$

- Diskretisierung der Zustände: 2 Belegung; 5 Beladung; 5 Außentemperatur

■ a – Aktionen (Ausgänge des EM)

- $a = [aHp, bHpSp1, bHpSp2, aSol, bSolSp1, bSolSp2, rSp1, rSp2]^T$

- Aktionen entsprechen einer Freigabe, die Schalthandlungen übernimmt die lokale Steuerung von Solarthermie oder Wärmepumpe aus;

aHp	Freigabe Wärmepumpe
$bHpSp1$	Wärmepumpe → Speicher 1
$bHpSp2$	Wärmepumpe → Speicher 2
$aSol$	Freigabe Solarthermie
$bSolSp1$	Solarthermie → Speicher 1
$bSolSp2$	Solarthermie → Speicher 2
$rSp1$	Entladung von Speicher 1
$rSp2$	Entladung von Speicher 2

RL Algorithmen

Belohnungsfunktion

- $R_1 = -(w_e E_{el} + w_1 D_{z1} + w_2 D_{z2})$, $w_{e,1,2}$: Gewichtungsfaktoren;
 E_{el} : Stromverbrauch
 D_z : Komfortverletzung (eine quadratische Form der Differenz e zwischen Ist- und Soll-Temperatur)

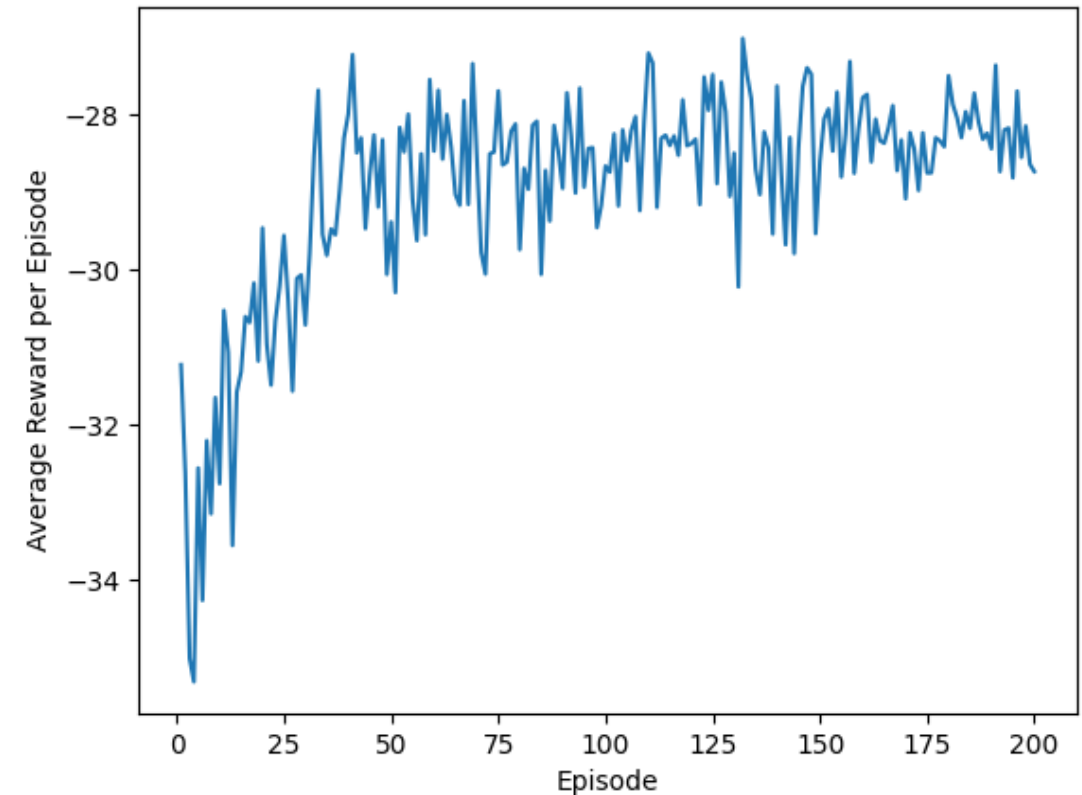
- $R_2 = -(w_e E_{el} + w_1 Cost_{T1} + w_2 Cost_{T2})$,

$$Cost_T = \begin{cases} 0, & e \leq 0 \\ -1, & 0 < e < 0.1 \\ -2, & 0.1 \leq e < 0.3 \\ -3, & 0.3 \leq e < 0.5 \\ -4, & 0.5 \leq e < 1 \\ -5, & 1 \leq e < 2 \\ -6, & 2 \leq e \end{cases}$$

Test mit zwei verschiedenen Reward Funktionen; quadratische Abweichung und empirische Funktion

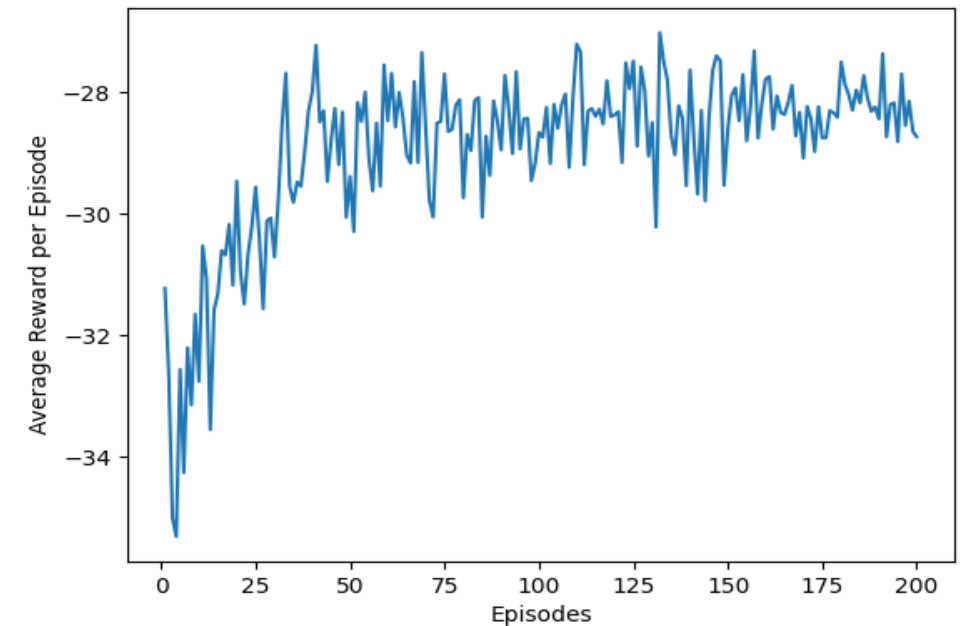
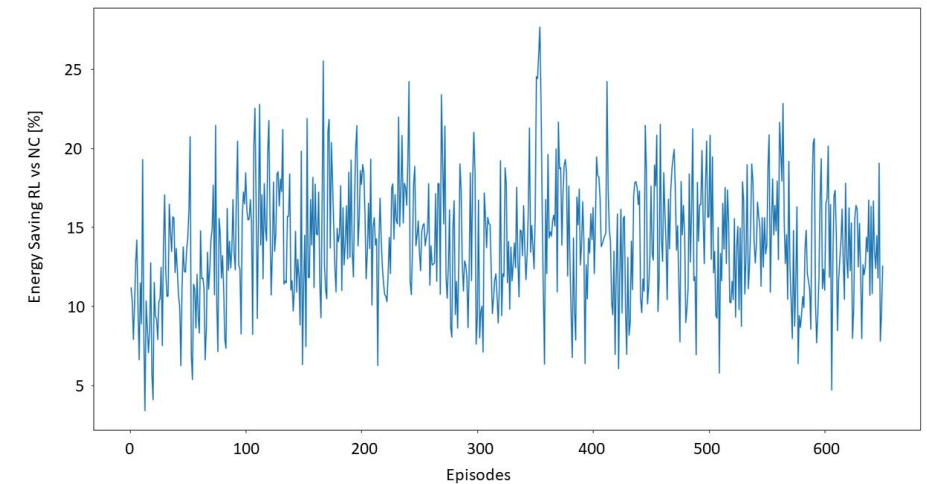
Reinforcement Learning als Energieverteilmanager - Ergebnisse

- Implementation
 - RL Agent als Python Programm
 - Energiesystem als FMU
- Training des RL-Agenten mit ausreichend vielen Episoden
- Ca. bis Episode 50 stetige Verbesserung (Lernen erfolgreich) der erzielten Energieeinsprung
- Betrachteter Zeitraum – 4 Wochen
 - Sim.Zeit ca. 1 Min pro Episode
 - Nach ca 1h ausreichende Ergebnisse



Durchschnittlicher Gewinn während Lernphase

- Absoluter Gewinn/Belohnung (R_2) für jede Episode während der Lernphase
- Lernvorgang nach ca. 50 bis 100 Episoden ausreichend
- ABER:
 - Welche Parameter führen zum besten Lernerfolg?
 - ϵ , γ , α – Parameter RL Agent
 - Gewichtungsfaktoren:
 $W_{electric} = 1$, $W_{z1} = 150$, $W_{z2} = 150$
 - Empirische Suche dauert lang und ist nicht unbedingt erfolgreich



Reinforcement Learning

Hyperparameter-Optimierung

- Im Bereich des maschinellen Lernens bezeichnet Hyperparameteroptimierung die Suche nach optimalen Hyperparametern. Ein Hyperparameter ist ein Parameter, der zur Steuerung des Trainingsalgorithmus verwendet wird.
- → automatisierte Suche nach den optimalen Hyperparameter ϵ, γ, α (Parameter RL Agent)
- Parametrierung des RL-Agenten
 $0 < \alpha < 1; 0 < \gamma < 1; 0 < \epsilon < 1$
- Optimale Einstellung für andere Parameter:
 - Gewichtungsfaktoren $w_{e,1,2}$ in der Belohnungsfunktion
 - initiale Werte vom Q-Table

Hyperparameter Optimierung mithilfe von *Nevergrad*

- Es entsteht ein multiobjektives Problem:
 - Maximierung des Gewinns aus der Belohnungsfunktion
 - Minimierung der Differenz der Gewinne z.B. in den letzten 50 Episoden
- Minimierung der Objektivfunktion (fitness function)
 $f = -\beta_1 \cdot \text{Gewinn} + \beta_2 \cdot \text{Error}$, $\beta_{1,2}$: Gewichtungsfaktoren
- unterschiedliche Optimierer z.B.: Random Search, TwoPointsDE, TBPSA
- Parametrierung
 $0 < \alpha < 1$; $0 < \gamma < 1$; $0 < \epsilon < 1$
 $w_e \in [1, 2, 3, 4, 5]$
 $w_{1,2} \in [100, 120, 150]$

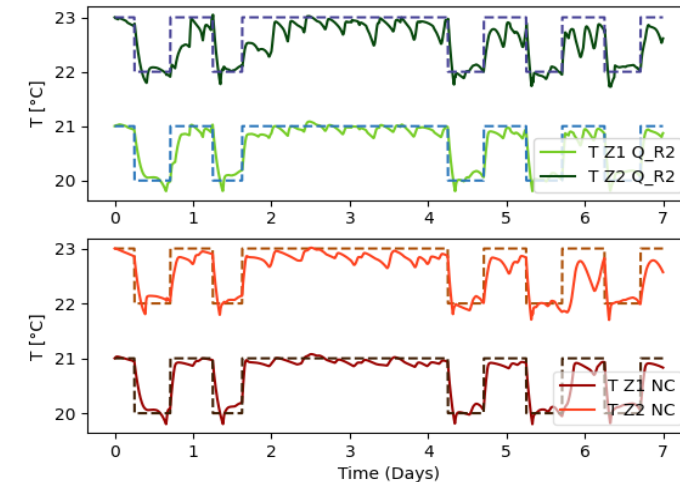
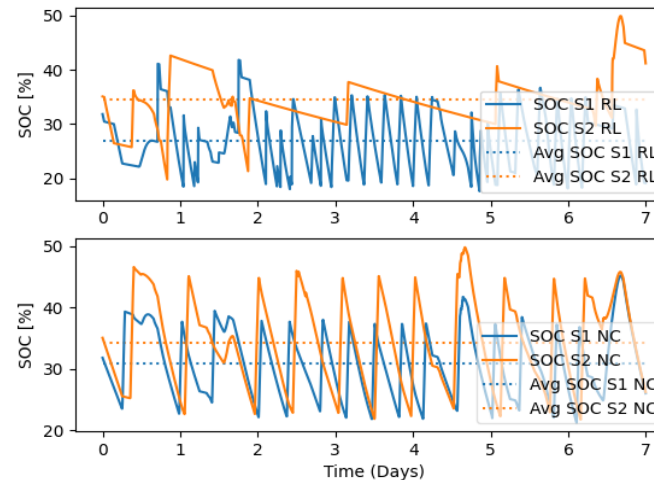
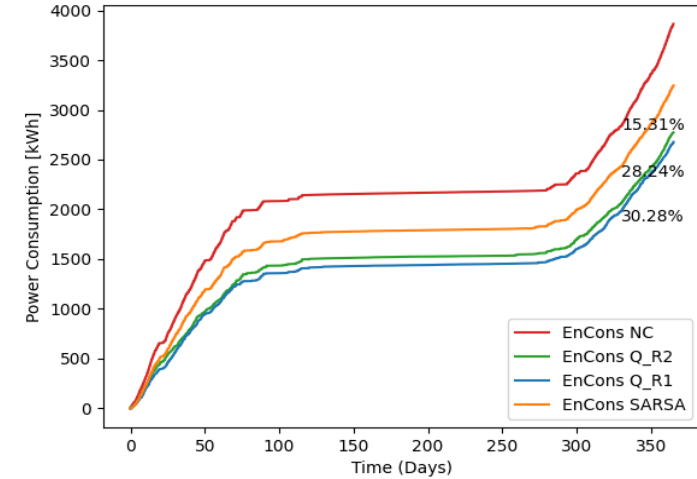
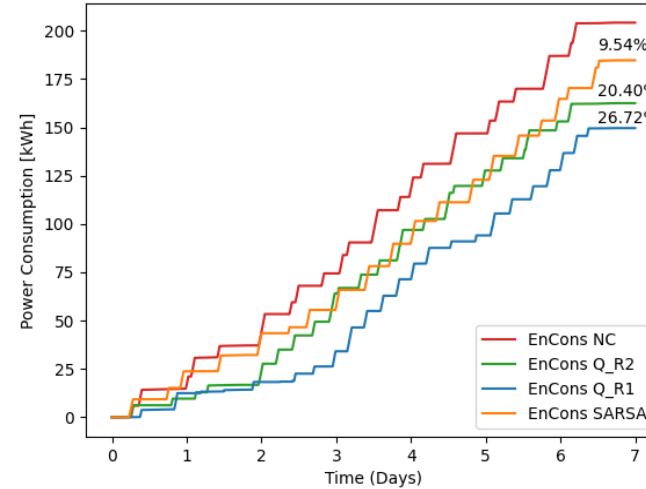
Hyperparameter Optimierung mithilfe von *Nevergrad*

- Verwendung des Frameworks Nevergrad
 - 5 Tage Rechenzeit zur Optimierung der Parameter mit je 50 Episoden je Parametersatz und 30 Tagen Simulationszeitraum
 - Resultate waren nicht besser als empirische, durch Probieren und Pi-mal-Daumen gefundene Hyperparameter, da starke Streuung
 - Sinnvoll für Neuronale Netze/DeepLearning Algorithmen, jedoch eher nicht für Q-Learning

	α	γ	ϵ	w_e	w_1	w_2
Random Search	0,0248	0,038	0,0996	5	100	100
TwoPointsDE	0,0445	0,036	0,5	4	100	100
TBPSA	0,115	0,083	0,048	4	100	100

Reinforcement Learning als Energiemanager - Ergebnisse

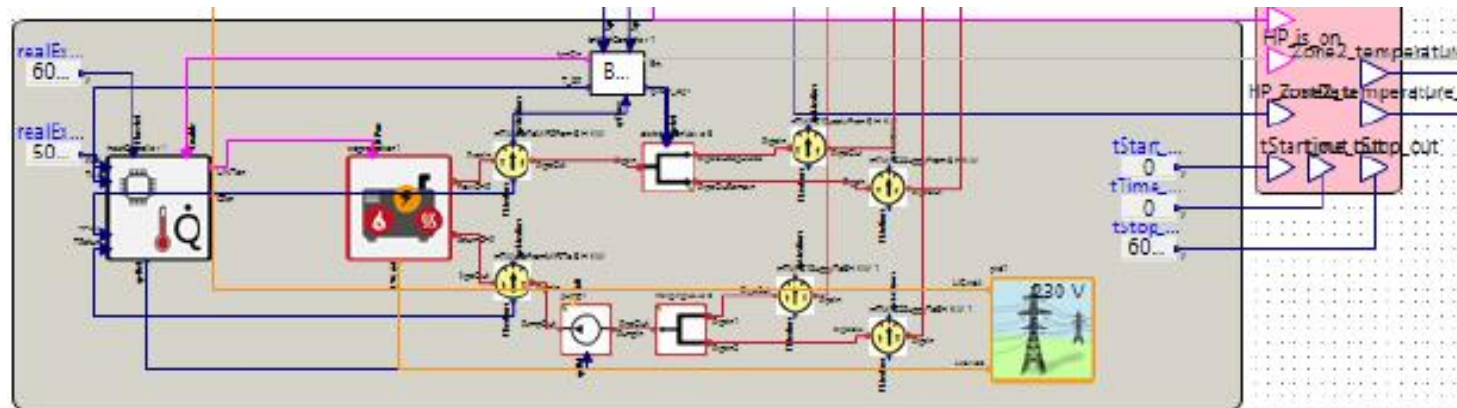
- Energieeinsparung 25 – 30 % ggü. einer „dummen“ Steuerung, bei der alle Aktionen/Freigaben immer gesetzt sind
 $a = [1,1,1,1,1,1,1,1]^T$
 - (abhängig von ϵ -greedy Faktor und Länge der Lernphase)
- Ersparnis vor allem durch geringere Beladung der Speicher
 - Mit RL: 28,5% und 31,6 %
 - Ohne RL: 31,6 % und 33,7 %
- Zonentemperaturen fast identisch
 - Abweichung: RL 0,02K tiefer
- Im Jahresverlauf bis zu 30% Einsparung



Comparison of RL (Reinforcement Learning) and NC (Normal Control):
 Upper left: Energy consum. in kWh per week, Upper right: Energy consum. in kWh per year,
 Lower left: State of charge in buffers 1 and 2 in %, Lower right: Zone temperatures in °C

Modellerweiterung mit BHKW (3. Wärmequelle)

- Ein wärmegeführtes Mikro-BHKW, Erdgas als Brennstoff → Strom für WP, Wärme für Gebäude
- Wärmepumpe (WP) Vermeidung hoher Stromkosten → Brennstoffkosten für BHKW geringer



Modellerweiterung mit BHKW

- Das Ziel der Reward-Function:
 - Garantie der Raumtemperatur in der Heizperiode
 - Kostenoptimale Lösung → Betriebskosten (Strom + Erdgas) niedrig zu halten
- $R = -(w_c Cost_{st} + w_f Fuel_{st} + w_1 D_{Z1} + w_2 D_{Z2})$ Dz: quad. Differenz e zwischen Ist- und Soll-Temperatur

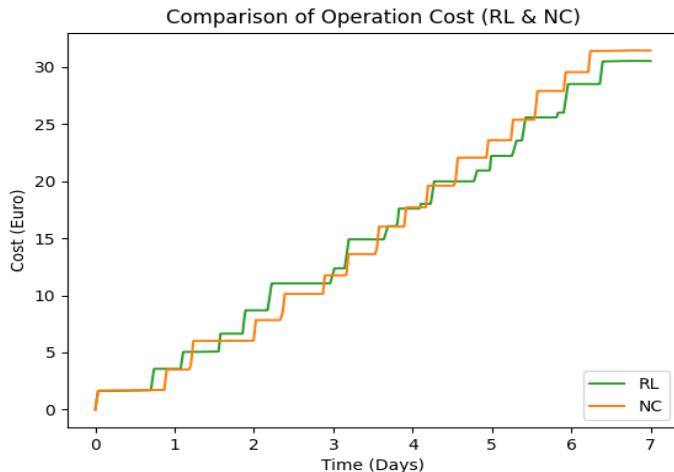
$$w_c := 12; w_f := 1; w_1 := 100; w_2 := 150$$

$Cost_{st}$: Betriebskosten; $Fuel_{st}$: Gasverbrauchs; D_Z : Komfortverletzung in Zone z1 und z2

$$Cost_{st} = \begin{cases} Erdgasverbrauch \times 6 \frac{Ct}{kWh} - Strom_{\ddot{u}} \times 11,8 \frac{Ct}{kWh} & Strom_{\ddot{u}} > 0, (Einspeisung) \\ Erdgasverbrauch \times 6 \frac{Ct}{kWh} - Strom_{\ddot{u}} \times 22,5 \frac{Ct}{kWh} & Strom_{\ddot{u}} < 0, (Strombezug) \end{cases}$$

wobei $Strom_{\ddot{u}} = Strom_{BHKW} - Strom_{WP}$.

Modellerweiterung mit BHKW



Vergleich Betriebskosten

Senkung der Betriebskosten um ca. 300€ durch Substitution des bezogenen Stromes durch Erdgas.

Vorher: ca. 4000 kWh Stromverbrauch

Nachher: 10390 kWh Gasverbrauch

Beachte: Kostenoptimierung, keine CO₂ Optimierung (!)

- 50% der Wärmeleistung der WP wird durch das BHKW ersetzt
- Strombedarf der WP wird komplett durch das BHKW abgedeckt
- Einspeisung von ca. 200 kWh in Netz
- Kosteneinsparung in Höhe von ca. 32% gegenüber dem Modell ohne BHKW
- Die Soll-Temperaturen werden erreicht; Komfort unverändert
- Die Betriebskosten sind leicht gesunken, aber Brennstoffverbrauch (in kWh) ist dabei gestiegen. WP ist effizienter, läuft aber weniger!
- Bewertung:
 - Der Agent hat gelernt, dass wenn er den Gasverbrauch erhöht und mehr Strom für die Einspeisung generiert, er die Betriebskosten senken kann.
 - Die Ursache kann der zu gering gewählte Gewichtungsfaktor w_f sein.
 - Freundlich fürs Portemonnaie, nicht für die Umwelt

Resultate - Reinforcement Learning

RL

%

25 - 30 % Einsparung

Vorteile

- Ressourcenbedarf im realen Betrieb gering, nur während der Lernens hoch (Simulation)
- Lernen des realen Betriebs danach weiter möglich (Adaption), allerdings mit geringerer Lernrate
- Ansatz ohne hohe Komplexität (geringer als MPC)
- Parametrierung über Hyperparameter

Nachteile

- Ergebnisse nicht immer optimal, da durch zufälliges Lernen gefunden
- Für komplexere Systeme sind Q-Tabellen nicht geeignet
- → Deep Neural Networks bzw Deep Reinforcement Learning Ansatz
- → Komplexer mit umfangreichen Lernvorgang

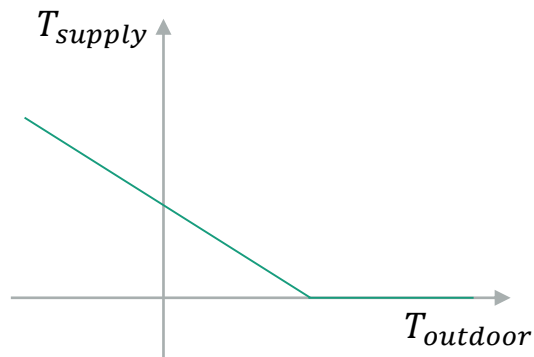
Zusammenfassung und Ausblick

- Zusammenfassung
 - RL hat Potential zum Einsatz als Energiemanager mit jeweiligen Vor- und Nachteilen
 - RL konnte in der Simulation das Einsparpotential zeigen
 - RL benötigt jedoch Initialaufwand (Modellbildung & Simulation) und einiges KnowHow im Bereich Machine Learning
 - RL bietet im Vergleich zu klassischer Regelung/MPC/Optimierung eine höhere Flexibilität, spez. in Bezug auf Rechenleistung und die benötigte Modellgenauigkeit, da es ein adaptives Verfahren ist
 - Für komplexere Aufgaben sind leistungsfähigere RL-Algorithmen notwendig
 - Mögliche Problematik der Rechenzeit

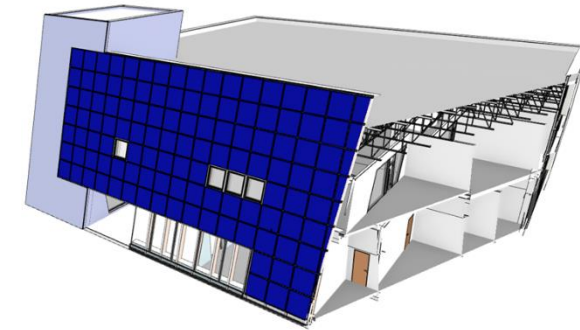
Vorlauftemperaturregelung

Klassische Vorlauftemperaturregelung:

- **Heizkurve** – Vorlauftemperatur abhängig der aktuellen Außentemperatur



- Moderne Gebäuden mit Niedertemperatur-Heizsystem: Träge gegenüber äußere Umwelteinflüsse und Innentemperatur stark von innere Wärmelasten abhängig



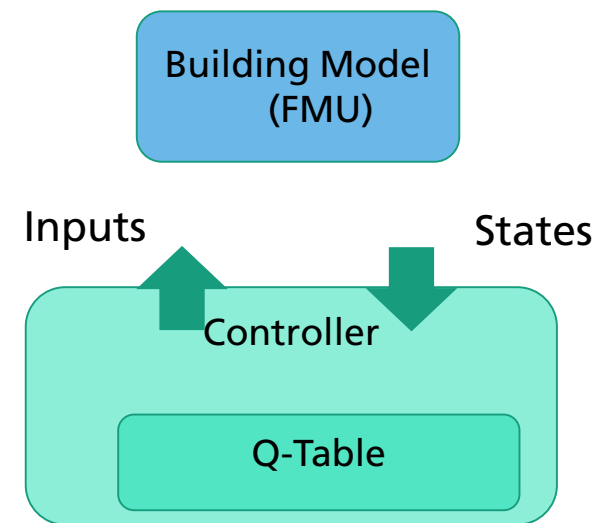
Demonstrator im Projekt Arche:

- FASA Bürogebäude in Chemnitz
 - Ausgestattet mit Fußbodenheizsystem + Heizkurve
- Probleme:
 - **Überhitzung** in Übergangszeit durch solarer Einstrahlung am Tag (Obergeschoss)
 - Unterkühlung am Montag durch die abgesenkte Solltemperatur am Wochenende
- Lösungsansatz: Reinforcement Learning für Optimierung der Heizkurve

Vorlauftemperaturregelung – Reinforcement Learning

- Konkrete Ansatz für FASA:
 - Action: Temperatur entspr. Heizkurve +
[+3K, +2K, +1.5K, +1K, +0.5K, 0K, -0.5K,
-1K, -1.5K, -2K, -2.5K, -3K, -3.5K, -4K, -5K]
 - Mögliche Zustände:
 - Aktuelle solarer Leistung
 - Aktuelle Stunde
 - Aktuelle Wochentag
 - Aktuelle Außentemperatur
 - Aktuelle gemittelte Zonentemperatur
 - Zukünftige solarer Gewinn
 - Zukünftige gemittelte Außentemperatur
 - Reward = Bestrafung bei Überheizung und Unterkühlung,
mit je eigener Wichtung

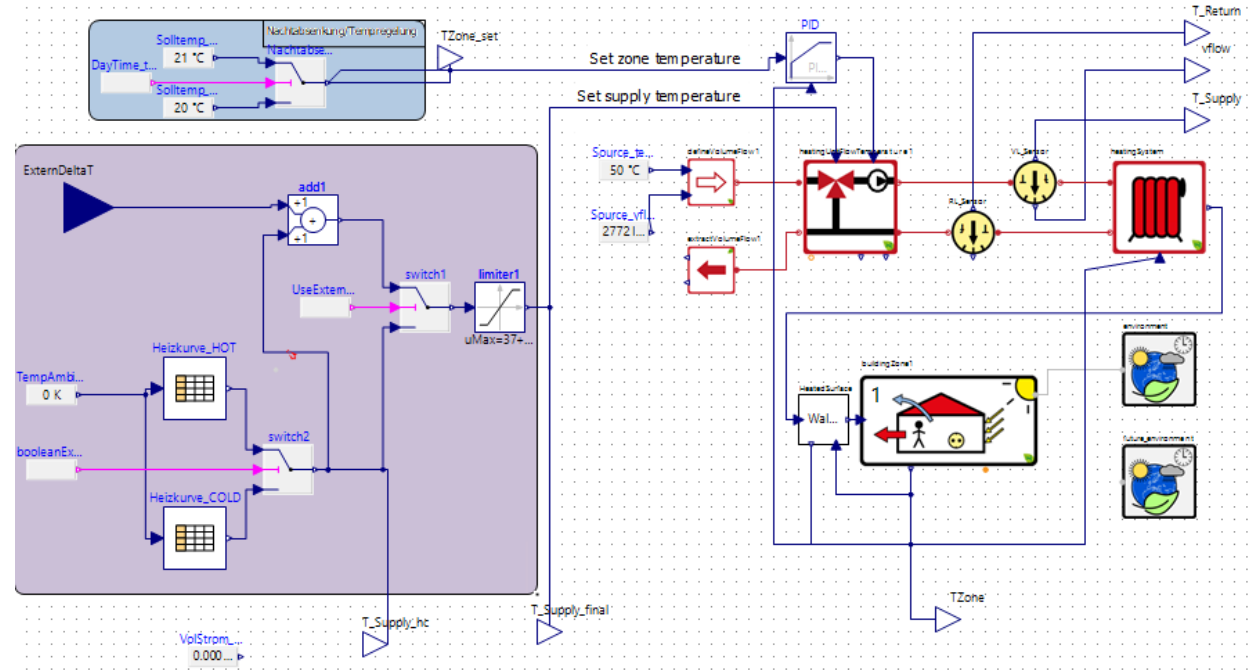
- Implementierung in Python (analog
Energieverteilmanager):



- Lege u.a. fest:
 - Time between action (TbA), Prädiktionszeit (PZ),
Wichtungen w-/w+, Lernepisoden

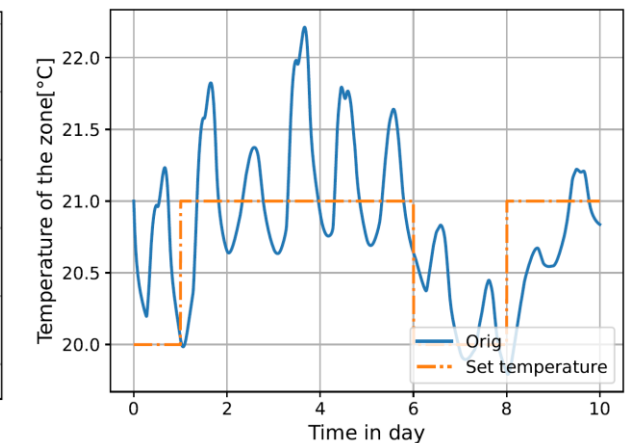
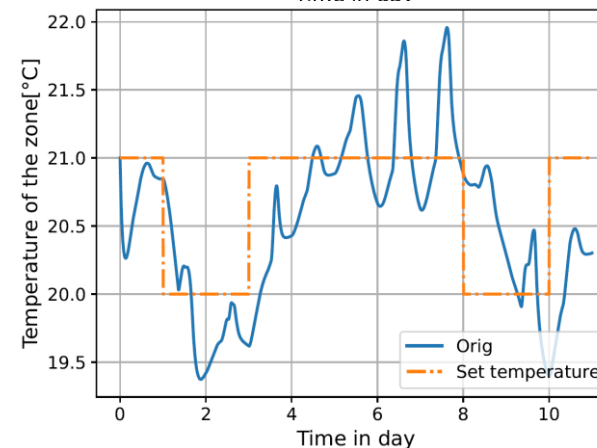
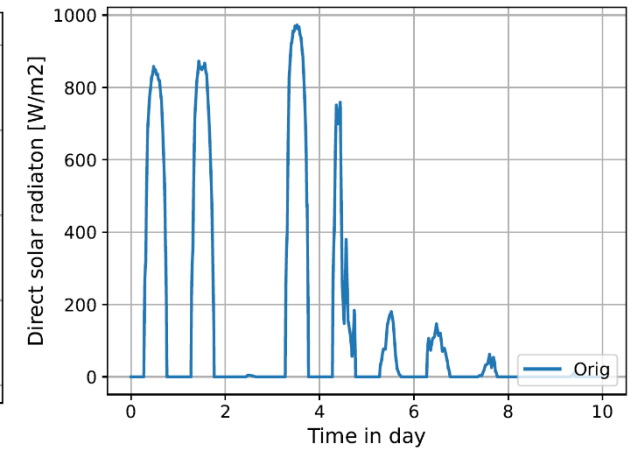
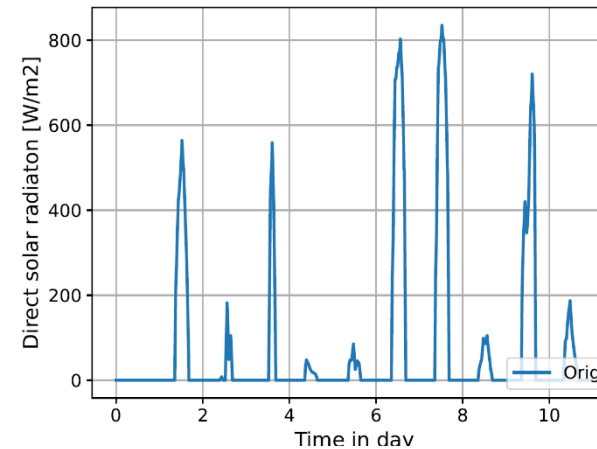
Vorlauftemperaturregelung – Reinforcement Learning

- Modell des Obergeschosses inkl. Heizungssystem zur Erzeugung von Lerndaten
- Approximation des Geschosses mit dem Modell einer Gebäudezone (900m²), teils an Messdaten angepasst.
- Fußbodenheizungssystem mit 45kW, PI Regler für den Erhalt der Zonentemperatur
- Vorlauftemperatur von der Heizkurve vorgegeben (normal und abgesenkter Betrieb), mit 37°C/23°C als Max/Min-Werte.
 - Abgesenkter Betrieb am Wochenende -> ebenfalls abgesenkter Sollwerte für Raumtemperatur
- Zusätzlicher Eingang für Anheben bzw. Absenken der Vorlauftemperatur



Vorlauftemperaturregelung – Reinforcement Learning

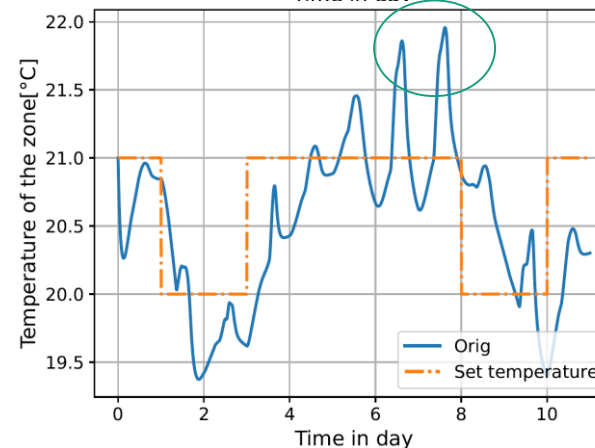
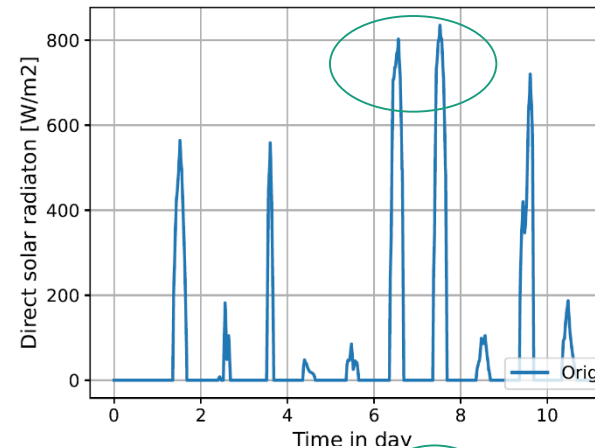
- Modell des Obergeschosses inkl. Heizungssystem zur Erzeugung von Lerndaten
- Approximation des Geschosses mit dem Modell einer Gebäudezone (900m²), teils an Messdaten angepasst.
- Fußbodenheizungssystem mit 45kW, PI Regler für den Erhalt der Zonentemperatur
- Vorlauftemperatur von der Heizkurve vorgegeben (normal und abgesenkter Betrieb), mit 37°C/23°C als Max/Min-Werte.
 - Abgesenkter Betrieb am Wochenende -> ebenfalls abgesenkter Sollwerte für Raumtemperatur
- Zusätzlicher Eingang für Anheben bzw. Absenken der Vorlauftemperatur



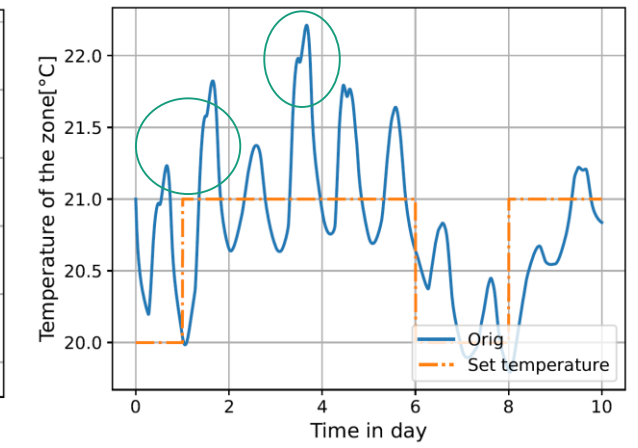
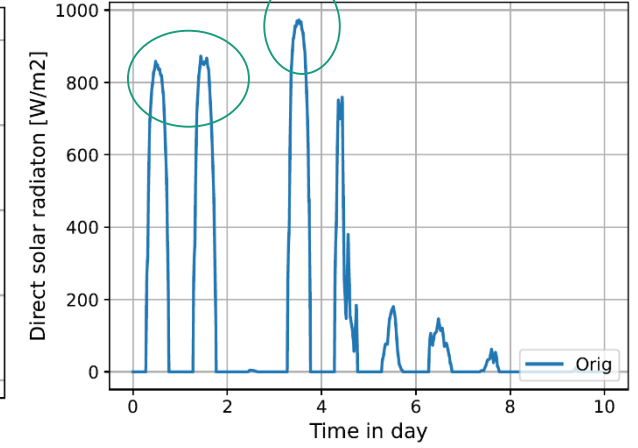
Vorlauftemperaturregelung – Reinforcement Learning

- Modell des Obergeschosses inkl. Heizungssystem zur Erzeugung von Lerndaten
- Approximation des Geschosses mit dem Modell einer Gebäudezone (900m²), teils an Messdaten angepasst.
- Fußbodenheizungssystem mit 45kW, PI Regler für den Erhalt der Zonentemperatur
- Vorlauftemperatur von der Heizkurve vorgegeben (normal und abgesenkter Betrieb), mit 37°C/23°C als Max/Min-Werte.
 - Abgesenkter Betrieb am Wochenende -> ebenfalls abgesenkter Sollwerte für Raumtemperatur
- Zusätzlicher Eingang für Anheben bzw. Absenken der Vorlauftemperatur

Szenario A

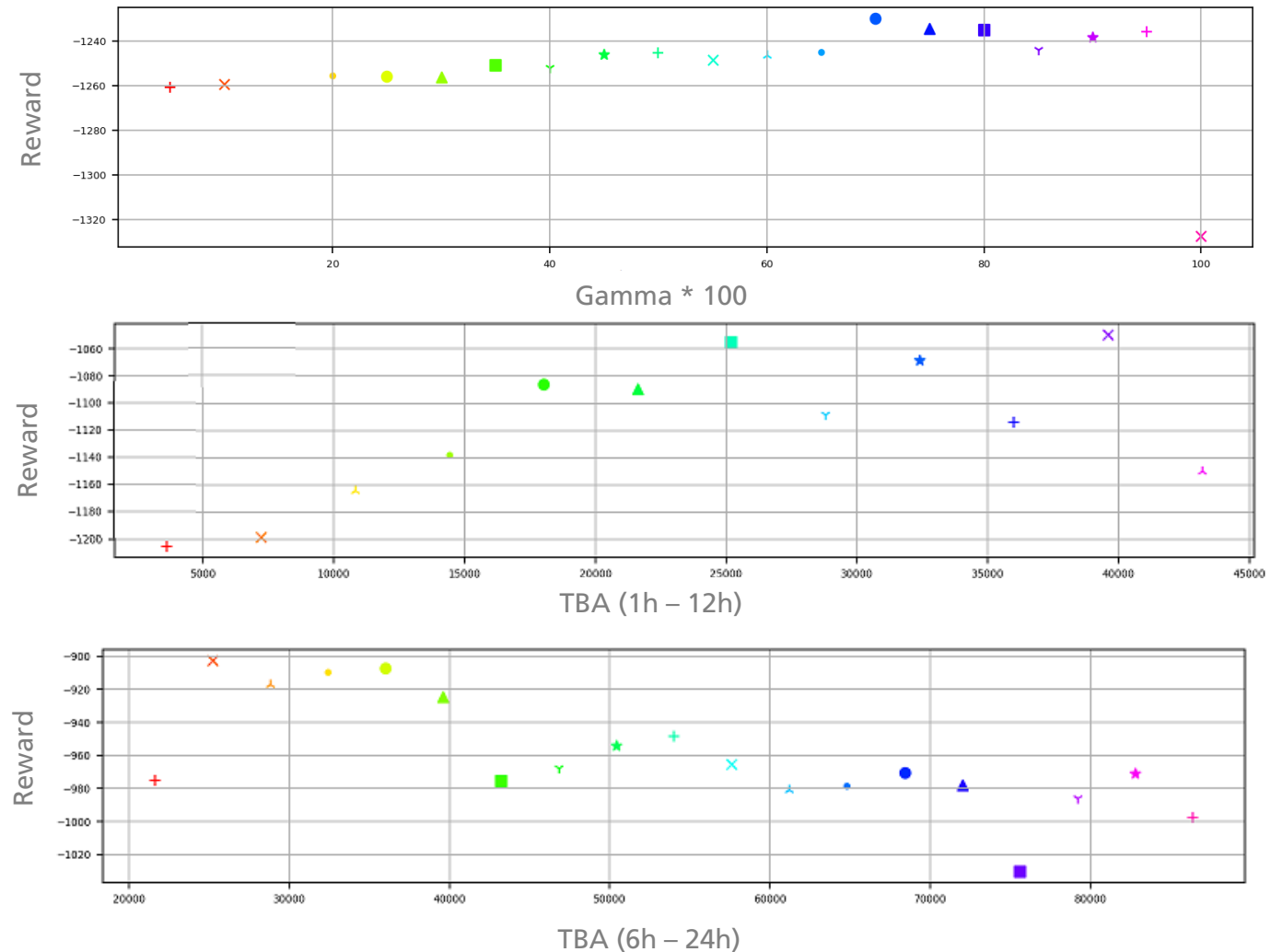


Szenario B



Vorlauftemperaturregelung – Reinforcement Learning

- Variation von Parameter
 - Q-Learning: Alpha, Epsilon, Gamma
 - Time between actions TBA, Prädiktionszeit
 - Wichtung $w+$ (zu warm) / $w-$ (zu kalt)
 - Anzahl der Lernepisoden
 - Änderung von Zuständen und deren Diskretisierung
 - Mit oder ohne aktueller solarer Strahlung
 - Mit oder ohne Zonentemperatur
 - Art der Diskretisierung der Wochentage
 - Art der Diskretisierung der Stunden
- Ziel: So wenig Zuständen wie möglich

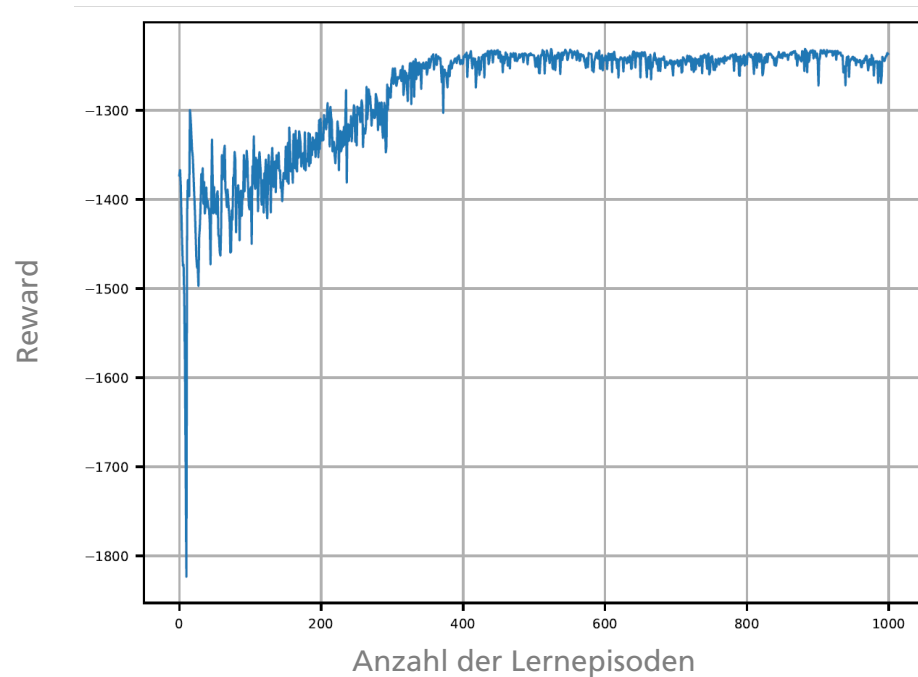


Vorlauftemperaturregelung – Reinforcement Learning

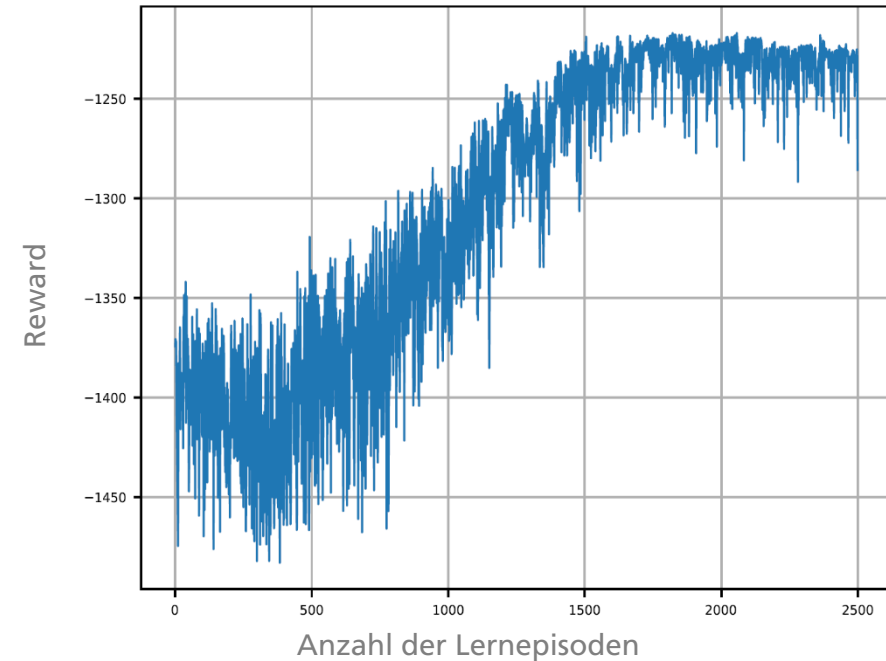
- Ergebnisse mit Einstellung:

- Lernzeitraum Jan-Apr (TRY); Wichtung $w_+/w_- = 0.7/0.5$; Prädiktionszeitraum 24h / **TBA 24h** (Action um Mitternacht)

Ohne Zonentemperatur

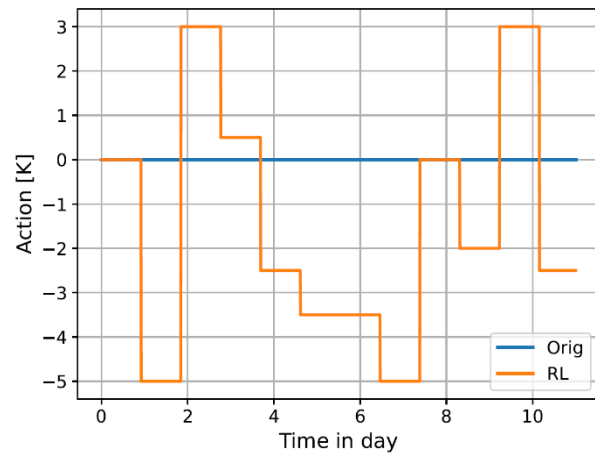
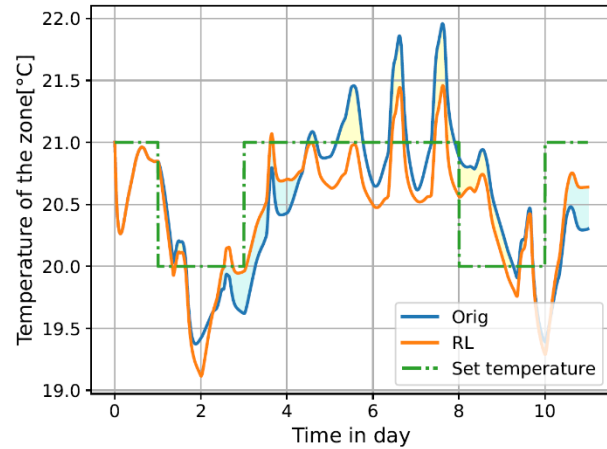


Mit Zonentemperatur

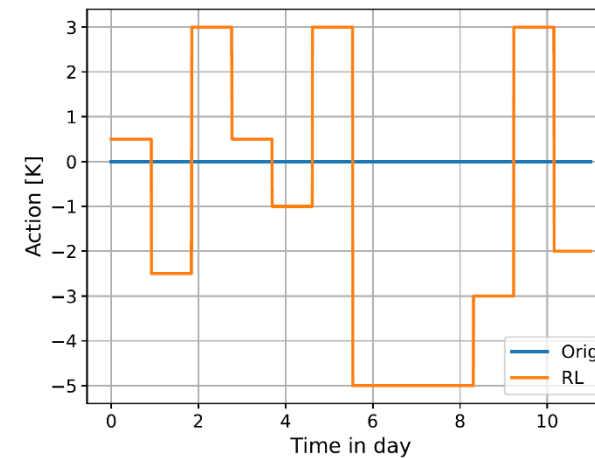
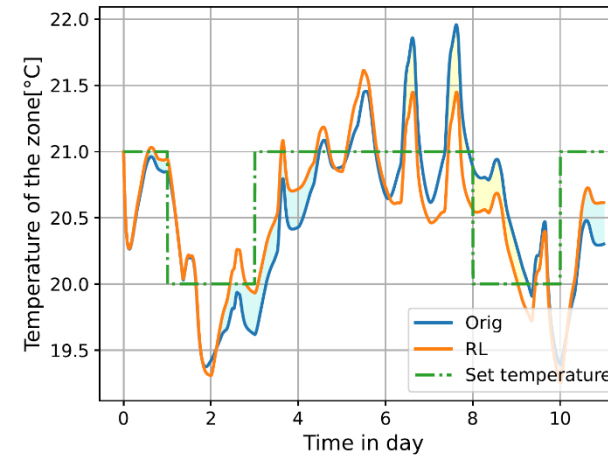


Vorlauftemperaturregelung – Reinforcement Learning

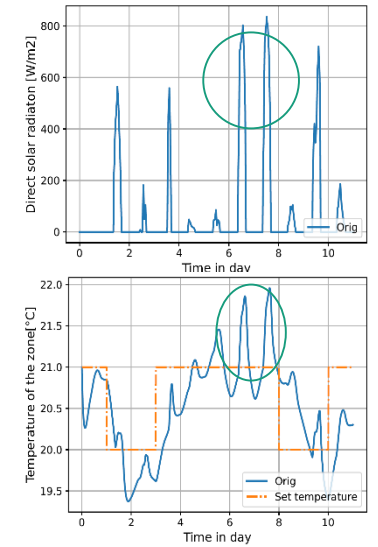
Ohne Zonentemperatur



Mit Zonentemperatur

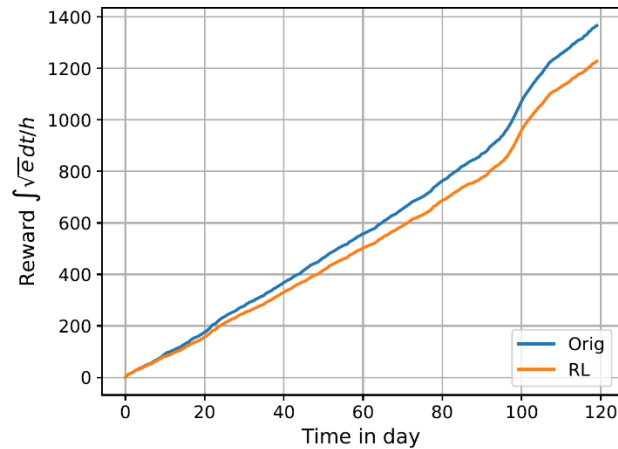
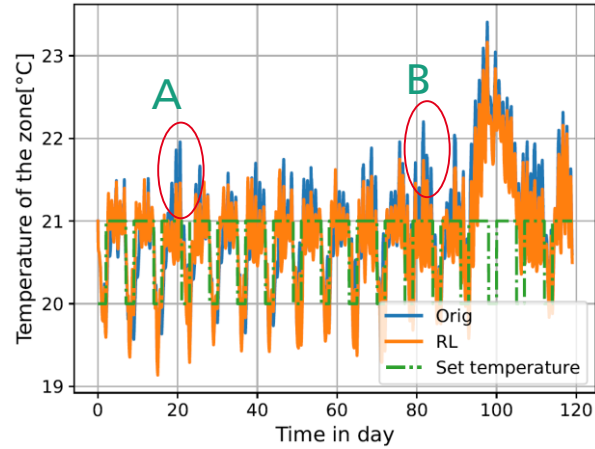


Szenario A

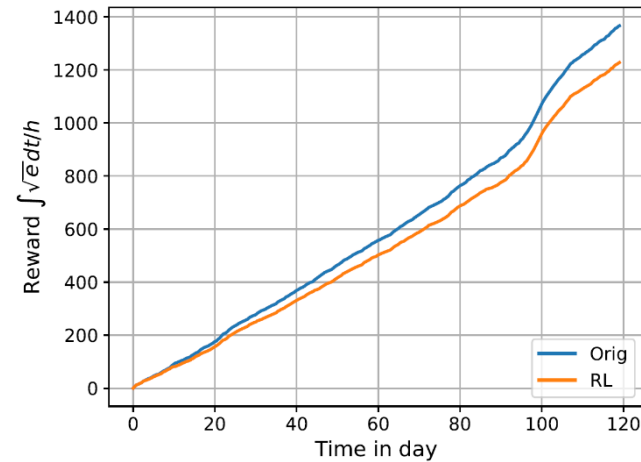
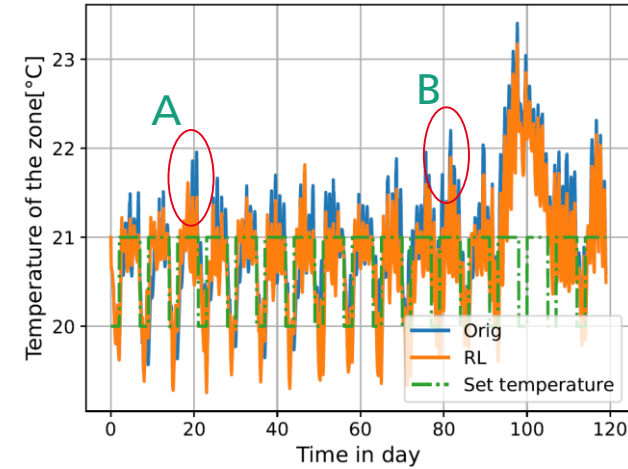


Vorlauftemperaturregelung – Reinforcement Learning

Ohne Zonentemperatur

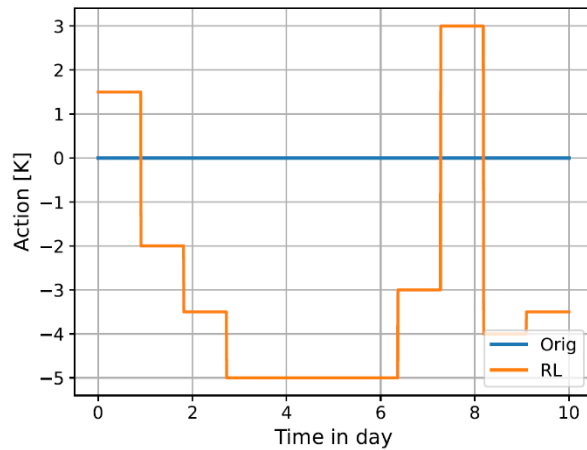
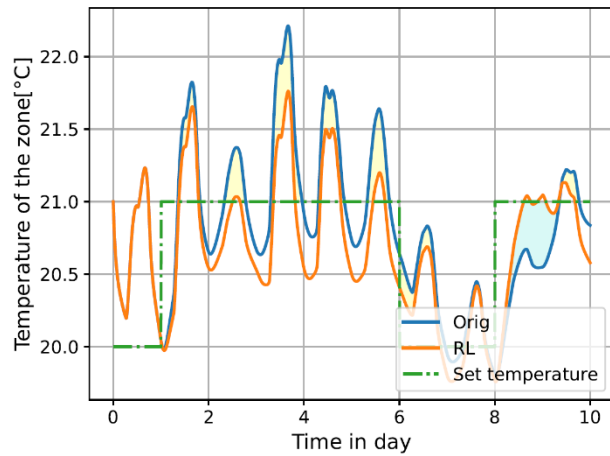


Mit Zonentemperatur



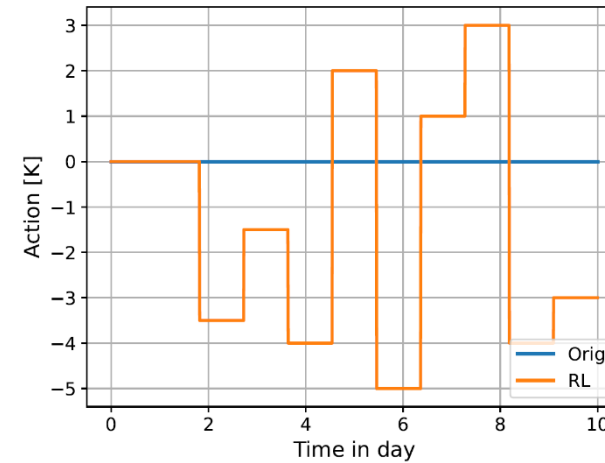
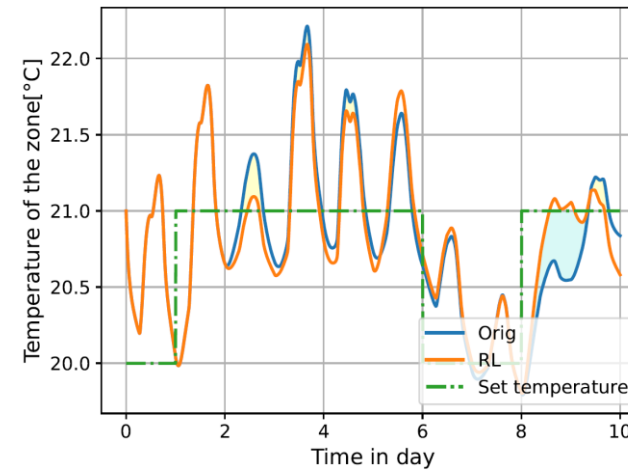
Vorlauftemperaturregelung – Reinforcement Learning

Ohne Zonentemperatur

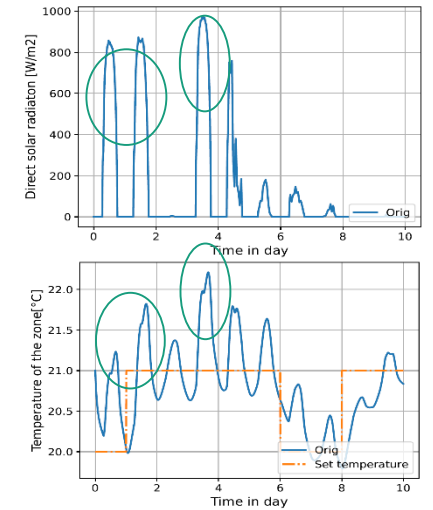


Andere Startwert für die Zonentemperatur

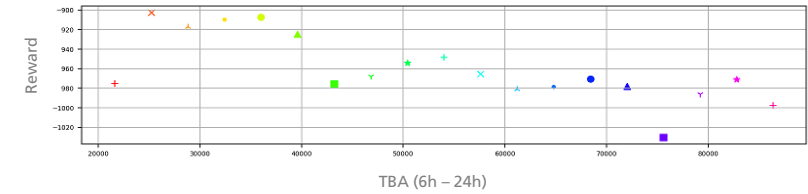
Mit Zonentemperatur



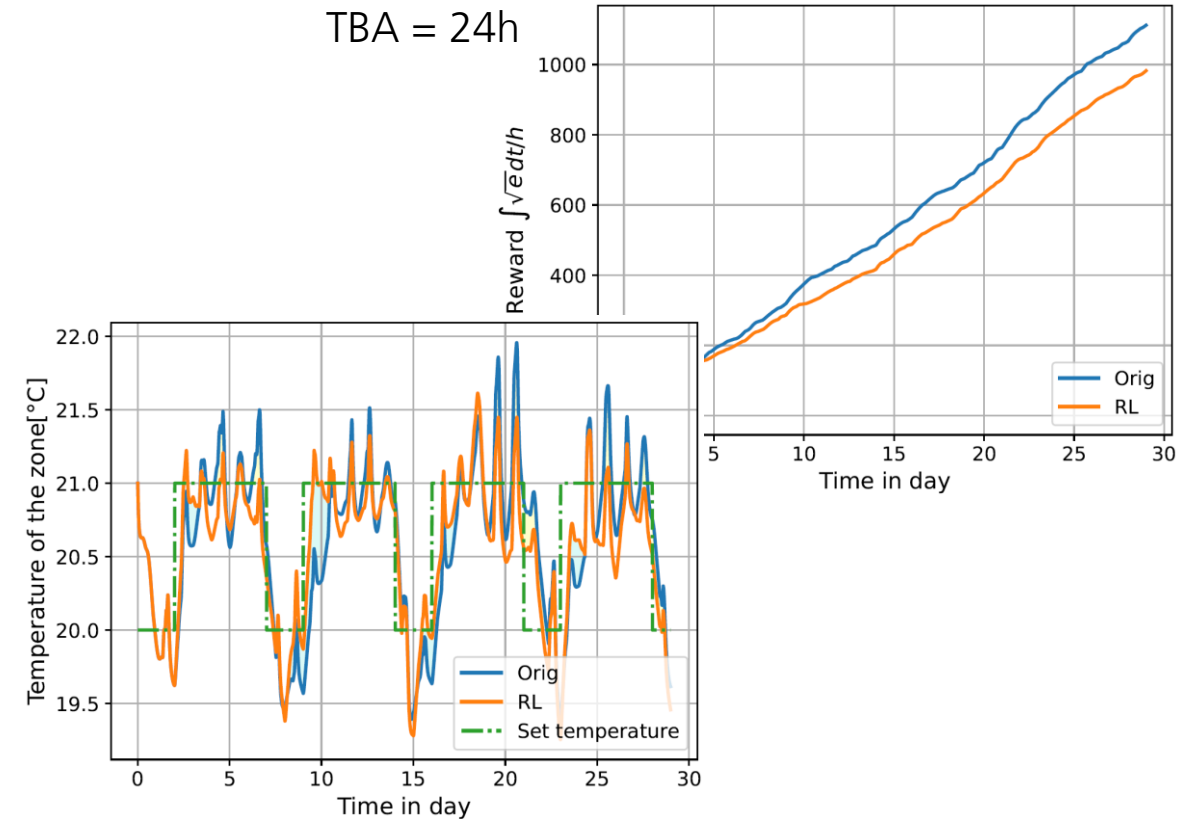
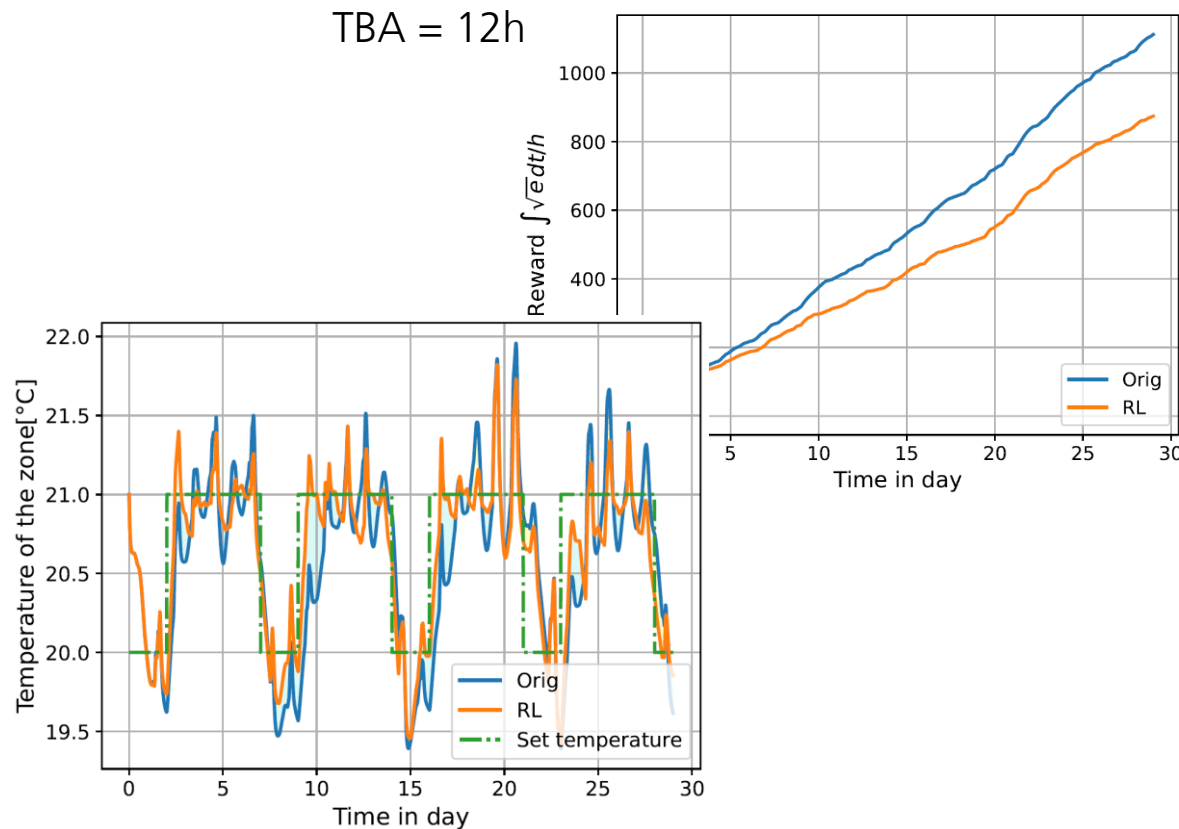
Szenario B



Vorlauftemperaturregelung – Reinforcement Learning

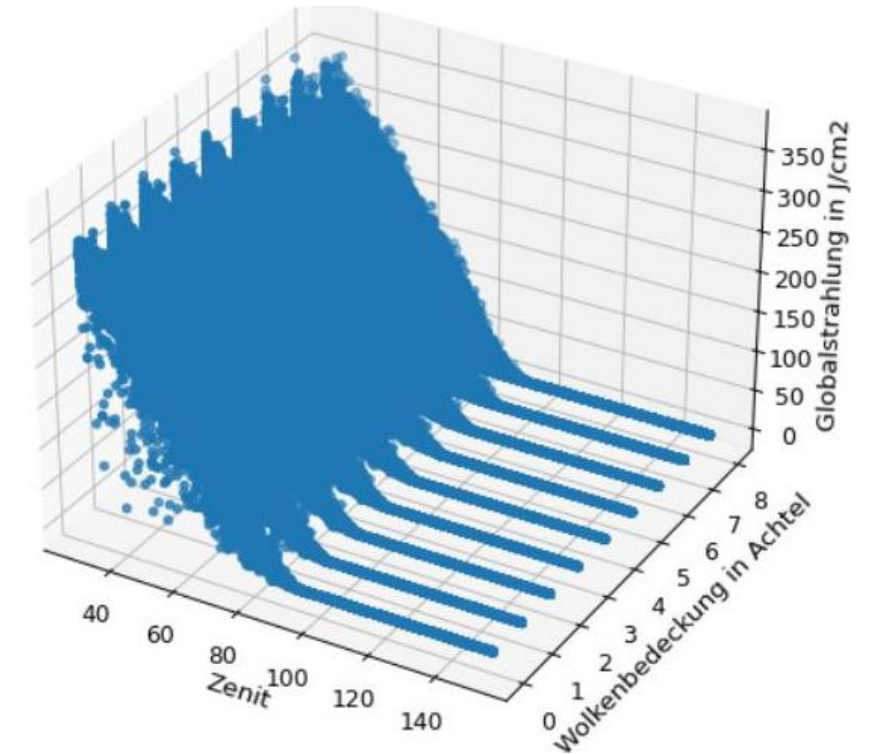


- Gesamtreward spiegelt nicht immer wider, wie gut der Regler mit spez. Situationen zurecht kommt:

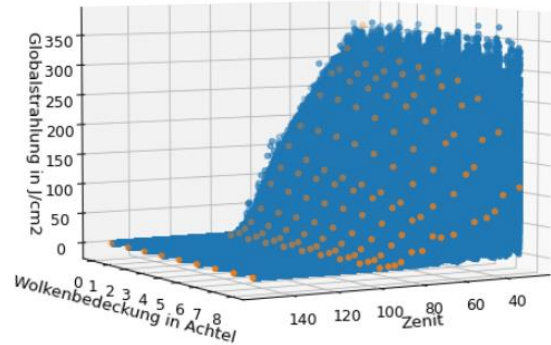


Vorlaufemperaturregelung – Prädiktion solarer Gewinn

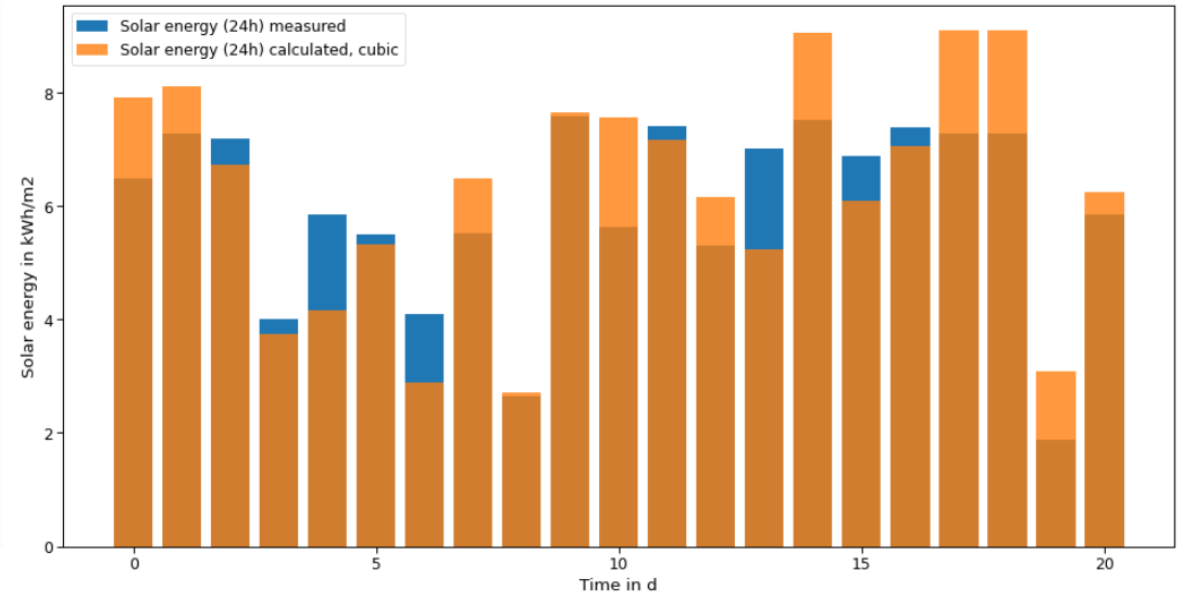
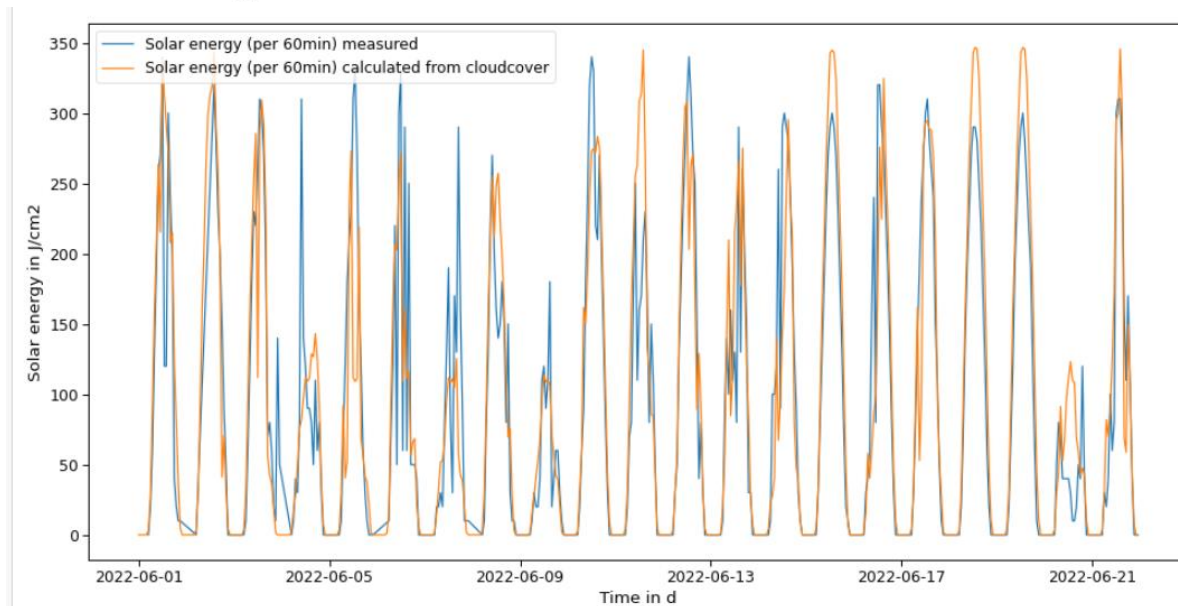
- Verfügbarkeit von Vorhersagen solarer Strahlung in Form von solarer Leistung oder solarer Energie in stündlicher Auflösung nicht immer (kostenlos) verfügbar
- Oft vorhanden:
 - Wolkenbedeckung in % (z.B. mithilfe Open Weather Map)
 - Sonnenscheindauer in min/h
- Ziel:
 - Berechnung der solaren Gewinne aus Wolkenbedeckung
- Ansatz:
 - Erstellung eines Kennlinienfelds aus historischen Daten (DWD: Chemnitz, 1981-2020)



Vorlauftemperaturregelung – Prädiktion solarer Gewinn



- Approximation der Messdaten mittels Polynome für ausgewählte Zenitwinkel (je nach Plausibilität der Datenlage)
- Aufspannen des Kennlinienfeldes durch die Polynome
- Validierung mithilfe von neuen Messdaten



Vorlauftemperaturregelung – Integration am realen Gebäude

- Integration der gelernten Q-Tabelle in das bereits vorhandene Expertensystem
 - Expertensystem: Regelbasiertes Handlungsempfehlungssystem
 - Output als Empfehlung an Gebäudebetreiber
 - Betreiber kann: Werte überprüfen → Übernehmen oder Ablehnen

→ Pausen Demonstrator

The screenshot displays the 'Fasa Building' management interface. The top navigation bar shows 'Floor: EG' and 'Discipline: ARK'. The left sidebar contains navigation options: User Services, Building entity, Building Monitoring, Spaces, Zones, HVAC systems, Occupancy prediction, and Timelapse. The main content area is titled 'Recommendations' and shows a list of suggestions for temperature adjustments. The third recommendation is highlighted with an orange border:

Today it is mild and cloudy. Maintain flow temperature. Clouds: 61% | Temp: 15.2°C
Space: Speicher;
23.9.2022, 12:05:02

The interface also shows a 3D architectural model of the building on the right, with a blue label '41 HOCHBAU' indicating a specific area. At the bottom right, there are navigation icons for hand, zoom, and scroll.

Vorlauftemperaturregelung – Integration am realen Gebäude

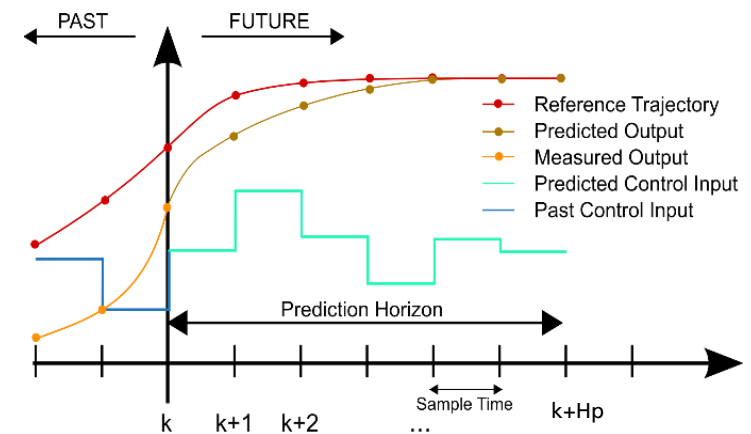
- Automatische Output der Empfehlung zu vorgegebenen Zeitpunkten
 - Mitternacht ungünstig für den Betreiber → Anlernen von Q-Tables zu anderen Zeitpunkten
- Spätere Integration des Reglers in das Gebäudeautomationssystem
- Fortführend:
 - Weiterentwicklung von komplexeren RL-Ansätzen

MODEL PREDICTIVE CONTROL (MPC)



Fabian Paschke (Fraunhofer IIS/EAS)

Chenzi Huang (Fraunhofer IIS/EAS)



Inhalt



- Einführung und Grundlagen
- MPC in der Gebäudetechnik
- Anwendung:
 - Mehrzonenregler für ein Hörsaalgebäude
 - Energiesystem mit Wärmepumpe und Speicher
 - Einzelraumregelung
- Zusammenfassung

Einführung und Grundlagen



- **Standardregler:** Stellgröße $u(t)$ wird aus Regelabweichung $e(t)$ berechnet, z. B. PID:

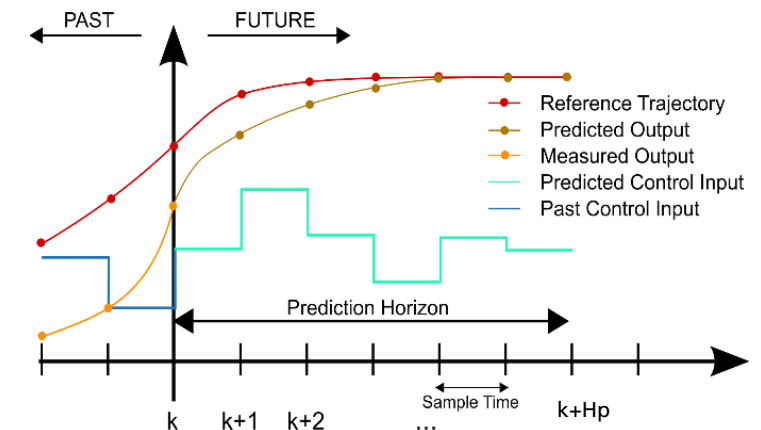
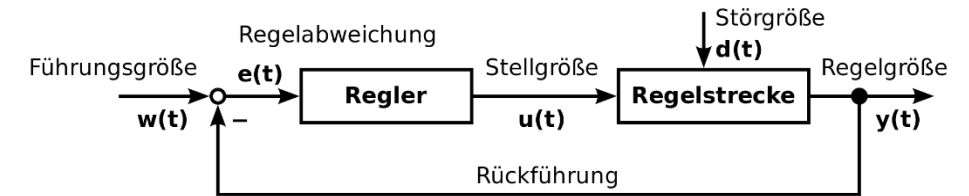
$$u(t) = k_P e(t) + k_I \int_0^t e(\tau) d\tau + k_D \frac{de(t)}{dt}$$

- **MPC:** Stellgrößenfolge $u(t)$ ($t = k, \dots, k + H_p$) wird durch Lösung eines Optimierungsproblems berechnet

$$u_{\text{opt}}(t) = \underset{u(t)}{\operatorname{argmin}} J(u(t), y(t))$$

$$\text{u. B. v. } u_{\min} \leq u(t) \leq u_{\max}$$
$$y_{\min} \leq y(t) \leq y_{\max}$$

- Erster Wert $u_{\text{opt}}(k)$ wird an Prozess ausgegeben
- Im nächsten Abtastschritt wird Optimierung unter Kenntnis neuer Prozessmessgrößen wiederholt



MPC in der Gebäudetechnik

Potentiale und Risiken von MPC



- **Vorteil:** Kosten- und/oder CO₂-Reduktion bei gleichbleibenden oder verbessertem Komfort (typisch: 5-50% je nach Anwendungsfall und Referenz)
- **Nachteil:** Deutlich höherer Entwurfsaufwand durch Modellbildung und Validierung
- ➔ MPC soll bzw. wird konventionelle Regelstrategien im Gebäudebereich nur dort ersetzen „wo es sich lohnt“
- Begünstigende Faktoren für Kosten-/ CO₂-Reduktion durch MPC in Gebäudeautomation:
 - „Große Anlagen“ mit hohen Energiekosten (bspw. Kälteverbundanlagen)
 - Thermische oder elektrische Speicher (explizit oder implizit)
 - Volatile Energiekosten oder eingeschränkt verfügbare Energiequellen (erneuerbare Energien)
 - Abhängigkeit der Effizienz eines Erzeugers von zeitlich variablen Umgebungs- oder Systemgrößen (Außentemperatur bei Luft-/Wasser-Wärmepumpe oder Rücklauftemperatur bei Brennwerttherme)
 - Signifikanter Hilfsenergieverbrauch mit ggf. nichtlinearen Abhängigkeiten zu variablen internen System oder Stellgrößen (Elektroenergieaufnahme der Lüfter in einer Lüftungsanlage)
- ➔ Zukünftig zunehmendes Potential in Gebäudeautomation

MPC in der Gebäudetechnik

Klassifikation



	Einzelraum	Mehrzonen	Energiesystem
Funktionsebene	Übergabe	Übergabe – Erzeugung	Erzeugung und Speicherung
Regelgrößen	Temperatur, rel. Feuchte und Luftqualität eines Raumes	wie bei Einzelraumreglern, jedoch mehrere Räume	Speicher-, Zulufttemperatur und/oder Vorlauftemperaturen der Heizkreise
Stellgrößen	Heiz- bzw. Kühlventilansteuerung, Zulufttemperatur, Verschattungsposition	wie bei Einzelraumreglern, jedoch zus. Zuluft-, oder Vorlauftemperaturen	Ansteuerung oder Vorlauftemp. der Erzeuger, Ventilpositionen und/oder Massenströme innerhalb der Energiesystems

- MPC – Gesamtgebäuderegler umfassen alle Funktionsebenen
 - ➔ Zerlegung des Optimierungsproblems in über- und untergeordnete Probleme
 - ➔ Sequentielle Lösung
- Weiterer Anwendungsbereich von MPC: Lokale Netzebene („Microgrids“)

Beispiel 1: Mehrzonenregler für ein Hörsaalgebäude

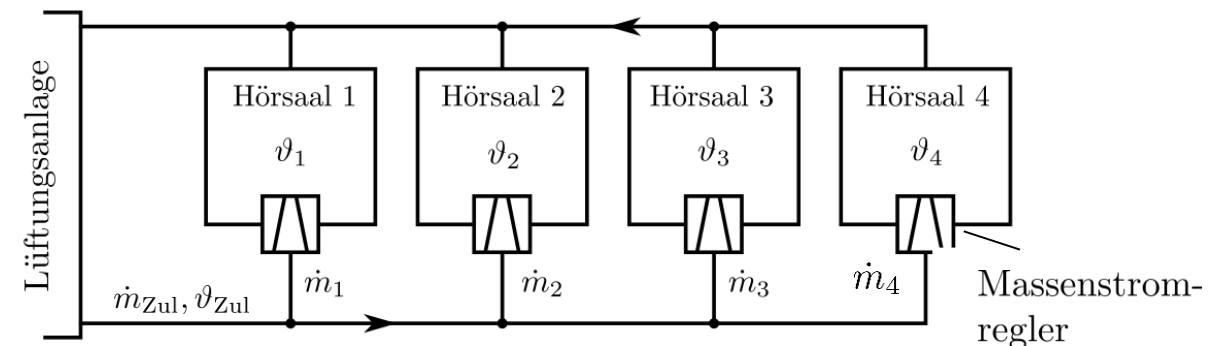
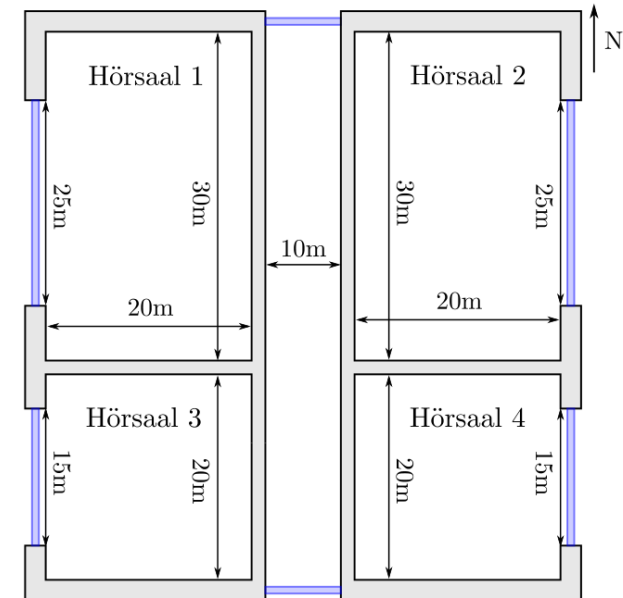
Gebäudeaufbau und Anlagentechnik

- Gebäude in Massivbauweise mit 4 Hörsälen und einem Flur
 - Wände aus Beton und Isolationsmaterial
 - Doppelglasfenster
- Heizen bzw. Kühlen über zentrale Lüftungsanlage mit Zulufttemperatur ϑ_{Zul}
- Raumtemperaturregelung in den Hörsälen über Massenströme

$$\dot{m}_i \quad (i = 1, \dots, 4)$$

Stell- und Regelgrößen:

- Regelgrößen: $\mathbf{y} := (\vartheta_1, \dots, \vartheta_4)^\top$
- Stellgrößen: $\mathbf{u} := (\vartheta_{Zul}, \dot{m}_1, \dots, \dot{m}_4)^\top$
- Störung: $\mathbf{d} := (\vartheta_{Aul}, \dot{Q}_{Sol})^\top$

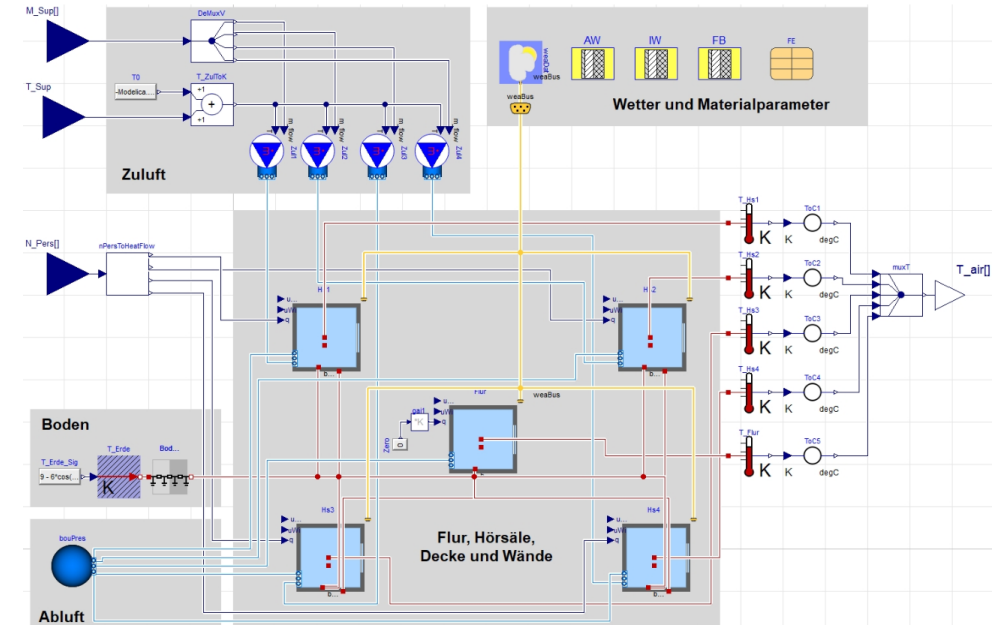


Beispiel 1: Mehrzonenregler für ein Hörsaalgebäude

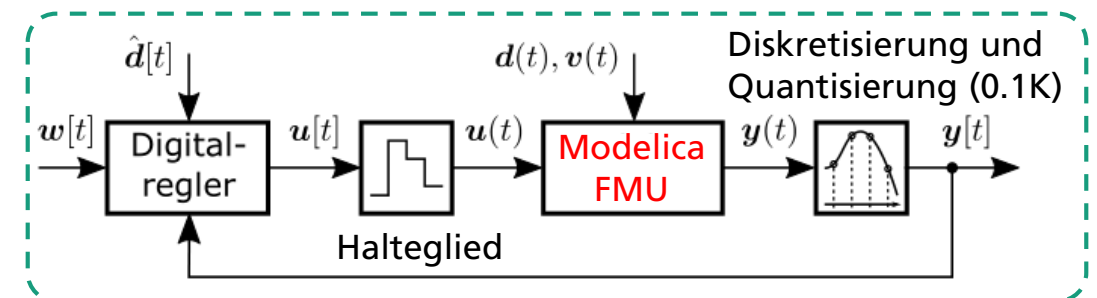
MODELICA Prozessmodell



- Implementierung des Gebäudemodells in MODELICA
- Modellierung mithilfe BUILDINGS Bibliothek
 - Koppelbares konzentriert parametrisches Einzelraummodell
 - Wärmeaustausch der Luft, Wände und Fenster durch Konvektion, Wärmeleitung, Infrarot- und Solarstrahlung
- Wetterdatensatz: Testreferenzjahr Berlin
- ➔ Nichtlineares DAE-System mit 137 Zustandsgrößen
- FMI-Export des MODELICA Modells
- ➔ Identifikation, Reglerentwurf und -validierung in MATLAB („Software in the Loop“)



MATLAB



Beispiel 1: Mehrzonenregler für ein Hörsaalgebäude

Regelungsziel und Referenzregelungsstrategie



- **Regelungsziel:** Hörsaaltemperaturen im Komfortband

$$\vartheta_{i,\min}[t] \leq \vartheta_i[t] \leq \vartheta_{i,\max}[t] \quad \forall i = 1, \dots, 4$$

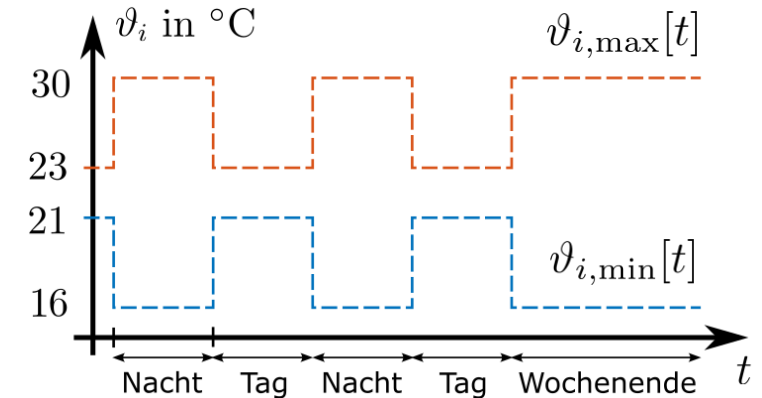
- **Referenzregelungsstrategie:**

- Obere/untere-Grenzwertregelung über Zuluftmassenströme $\dot{m}_i[t]$ durch PI-Regler
- Zulufttemperatur $\vartheta_{Zul}[t]$ über Zweipunktregler

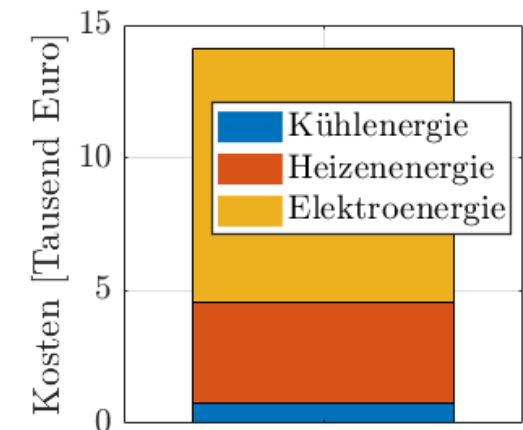
- **Simulation:** Regelungsziel erfüllt, aber hohe Elektroenergiekosten durch quadratische Massenstromabhängigkeit der Lüfter:

$$P_{El}[t] = \kappa_{El} \dot{m}_{Zul}^2[t] \quad (\dot{m}_{Zul}[t] = \sum_{i=1}^4 \dot{m}_i[t]) \quad \kappa_{El} \dots \text{Parameter}$$

➔ Ziel: Reduktion der Gesamtenergiekosten durch MPC



Jahresenergiekosten



Beispiel 1: Mehrzonenregler für ein Hörsaalgebäude

MPC Modell und Optimierungsproblem



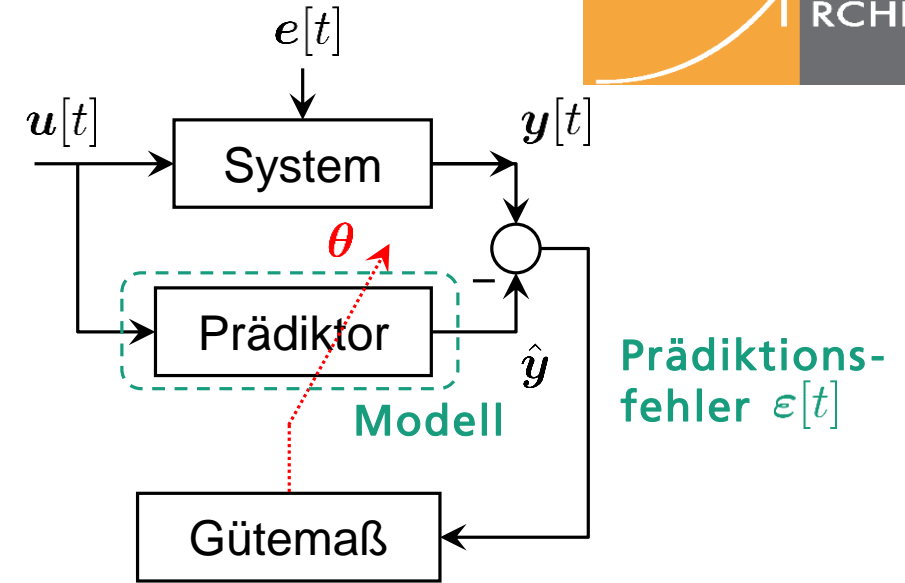
- Reduzierte Modelle für MPC mittels Systemidentifikation:
 - Daten durch Simulation des MODELICA Prozessmodells über 12 Wochen
 - Modellidentifikation mittels Prädiktionsfehlerminimierung
 - 4 Einzelraummodelle (Wärmeaustausch zw. Hörsälen wird nicht berücksichtigt)
 - Verkoppeltes Gesamtgebäudemodell

■ Kostenfunktion: Minimierung der Betriebskosten

- Heiz- und Kühlenergiekosten
- Elektroenergiekosten für Lüftungsanlage

■ Nebenbedingungen: Hörsaaltemperaturen im Komfortband

➔ NLP mit 484 Variablen, 964 linearen und 768 nichtlinearen NB



T_A	Abtastzeit	1h
k	Prädiktionshorizont	96h
w_{HK}	Heiz-/Kühlkosten	0.1€/kWh
w_{El}	Stromkosten	0.3€/kWh
η_{Wrg}	Effizienz Wärmerückgewinnung	0.75
κ_{El}	Parameter Lüfter	0.2kW/(kg/s) ²
c_L	Wärmekap. Luft	1.005kJ/kgK

Beispiel 1: Mehrzonenregler für ein Hörsaalgebäude

Simulationsergebnisse

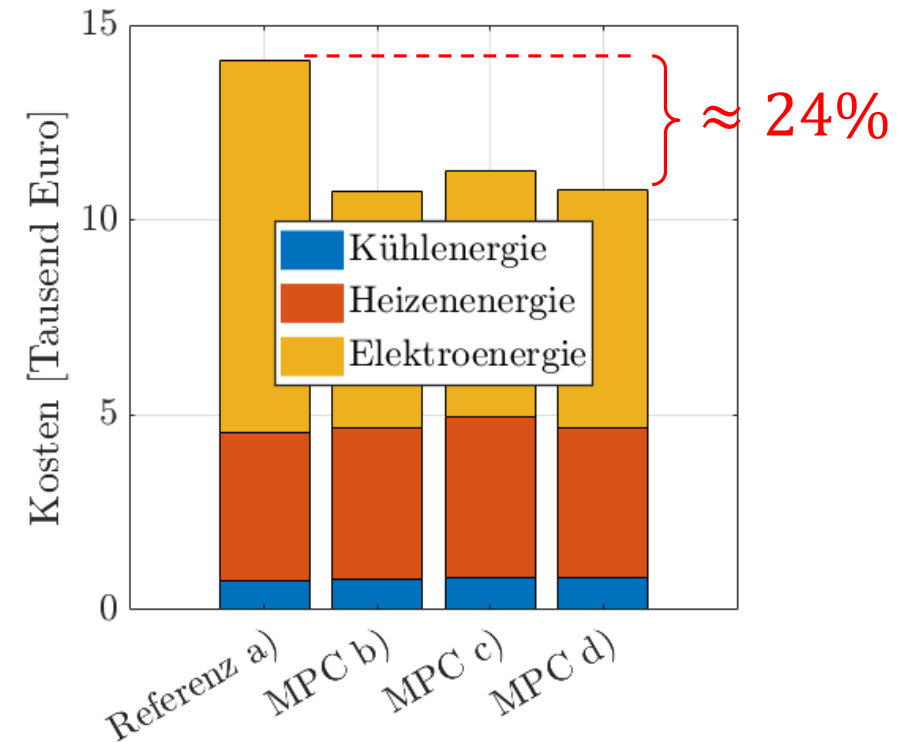


Jahressimulation für die 4 Fälle:

- Nur Referenzregelung (ohne MPC)
- MPC mit Einzelraum Modellen (ideale Wettervorhersage)
- MPC mit verkoppeltem Gesamtgebäudemodell (ideale Wettervorhersage)
- Wie unter b) nur mit einfachen Wettermodell für Außentemperatur und Globalstrahlung

Ergebnisse:

- $\approx 24\%$ Reduktion der Energiekosten durch Vorkühlen bzw. -heizen des Gebäudes mittels MPC
- Etwas höhere Gesamtkosten mit (genauerem) Gesamtgebäudemodell (vermutlich aufgrund "schwierigerem" NLP)
- Ungenaue Wettervorhersage hat vernachlässigbaren Einfluss



Beispiel 2: Energiesystem mit Wärmepumpe und Speicher

Regel- und Stellgrößen



- **Regelgröße:** Vorlauftemperatur am Heizkreisverteiler $T_{\text{HKV,VL}}(t)$

- Sollwert wird über Heizkurve(n) vorgegeben

$$T_{\text{HKV,VL}}(t) \geq \max_i T_{\text{HKi,VL,Soll}}(T_A(t))$$

- Vorlauftemperaturregelung der Heizkreise bleibt intakt → Prinzip kann auf andere Gebäude leicht übertragen werden

- **Stellgrößen:**

- Wärmepumpenmodulation M_{WP}
- Kesselmodulation M_{K}

- **Eingangsrößen:**

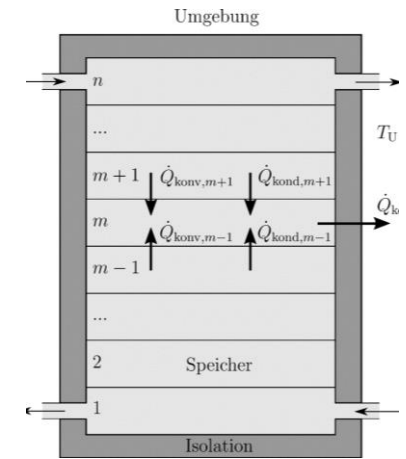
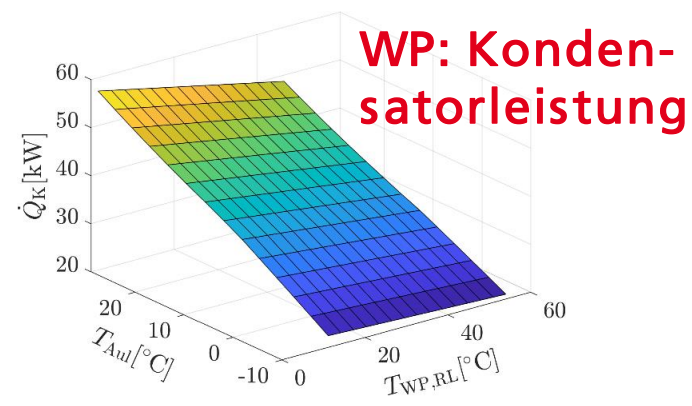
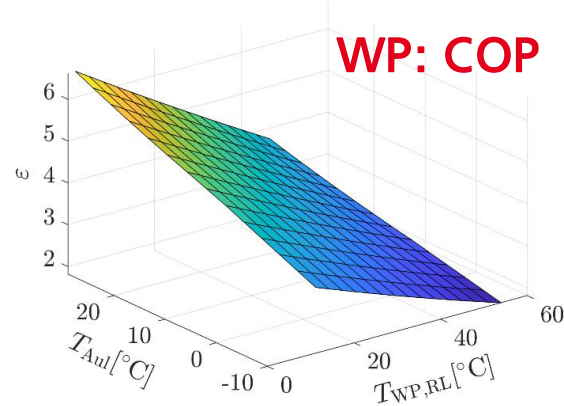
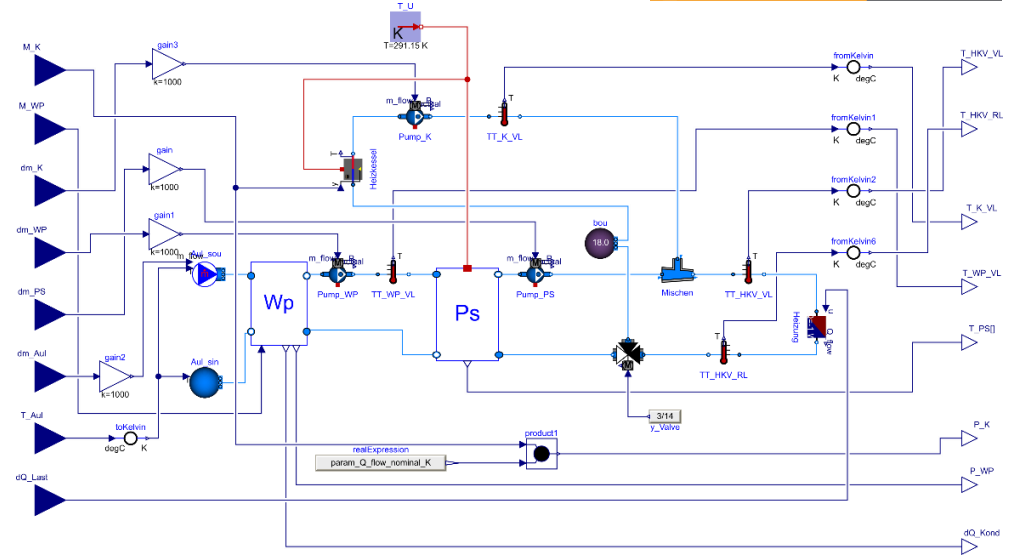
- Außentemperaturvorhersage $T_A(t)$
- Vorhersage der Heizlast $\dot{Q}_L(t)$

Beispiel 2: Energiesystem mit Wärmepumpe und Speicher

Modellbildung



- Implementierung des Prozessmodells in MODELICA (BUILDINGS Library)
- FMI-Export des MODELICA Models → Simulation des Regelkreises in MATLAB (analog zum Beispiel 1)
- MPC Modelle:
 - Kennlinienmodell für Gasbrennwertkessel (statisch)
 - 2 Kennfelder für Wärmepumpe
 - Schichtladespeichermodell für Pufferspeicher (dynamisch)



Beispiel 2: Energiesystem mit Wärmepumpe und Speicher

Regelungsstrategien



Referenzregelung:

- Ansteuerung der Wärmepumpe über Zweipunktregler mit Hysterese
- Betrieb der WP nur wenn COP ausreichend hoch
- PI-Regler für Gasbrennwertkessel

	Kosten 2019	Kosten Annahme
Gaspreis	6.3 ct/kWh	10 ct/kWh
Strompreis	22.5 ct/kWh	22.5 ct/kWh
COP _{min}	3.55	2.25

MPC-Regler:

- Abtastzeit $T_S = 1h$, Steuer und Prädiktionshorizont $H_p = 24h$
- Minimierung der Gas- und Stromkosten

$$J = \sum_{k=0}^{H_p} K_{\text{Strom}} P_{\text{WP}}[k] T_S + \sum_{k=0}^{H_p} K_{\text{Gas}} P_{\text{K}}[k] T_S \cdot$$

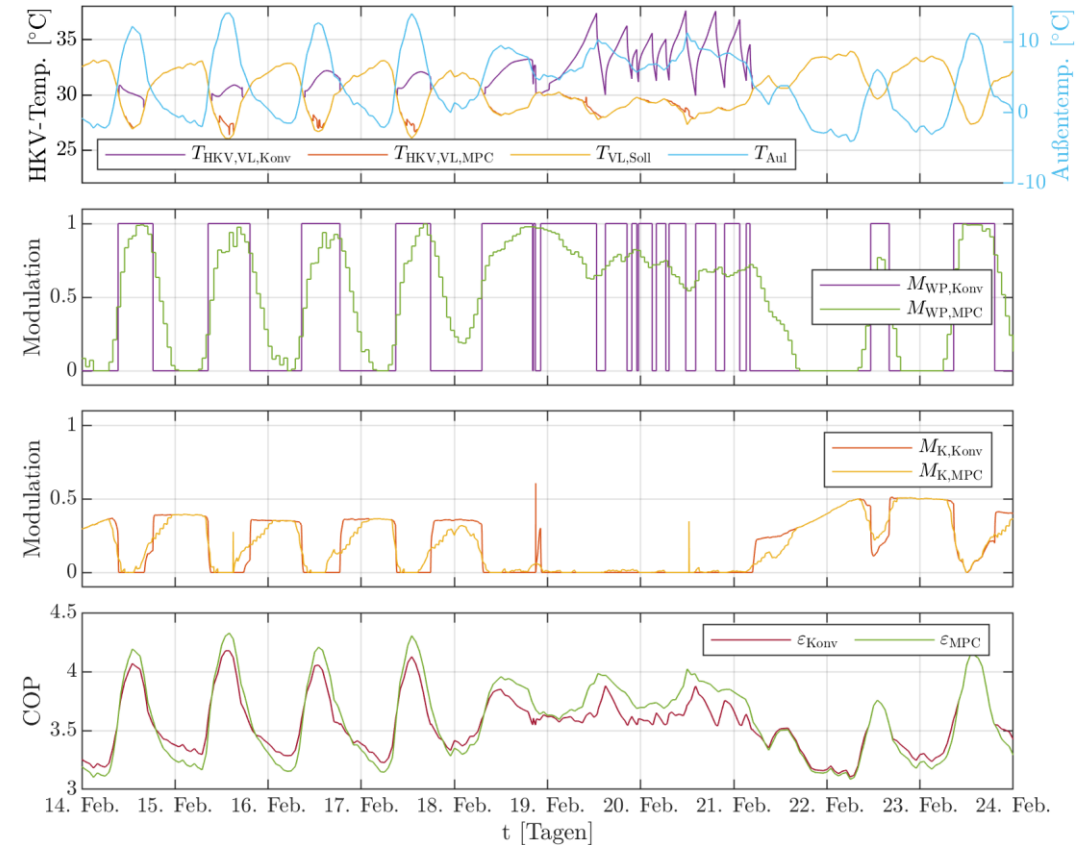
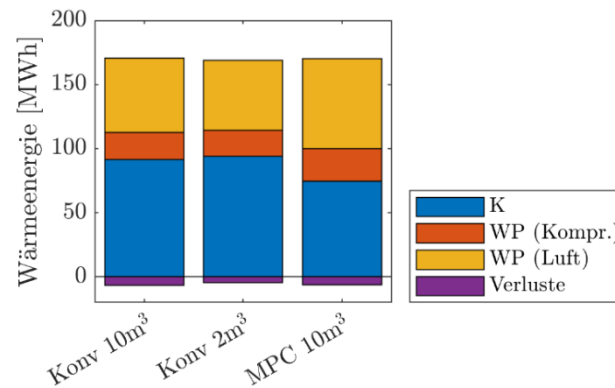
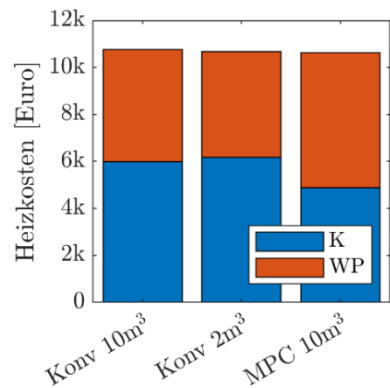
- In beiden Fällen: Fluidpumpen AN wenn WP bzw. Kessel AN (Massenstrom = nom. Massenstrom der Pumpen)

Beispiel 2: Energiesystem mit Wärmepumpe und Speicher

Simulationsergebnisse



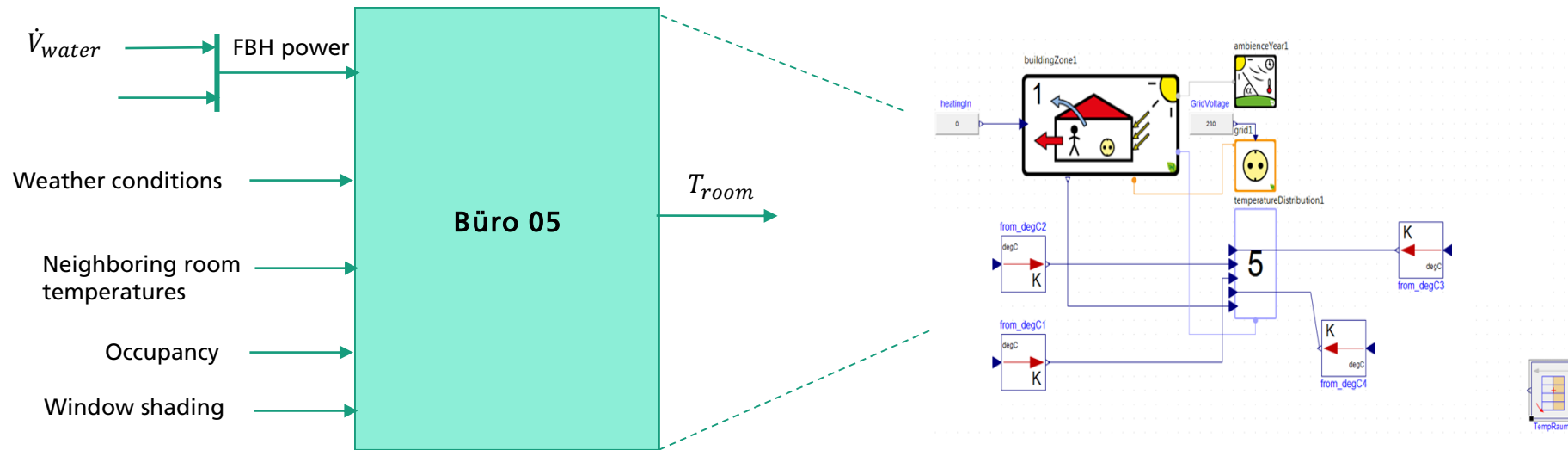
- Heizkostenvergleich der Jahressimulation
- Unwesentliche Kostenminimierung durch MPC
 - Grund: ähnliche Betriebsführung der beider Regelstrategien
- Wesentlicher Unterschied: MPC verstärkt Nutzung der Wärmepumpe



Beispiel 3: Einzelraumregelung

Regelungsstrategie

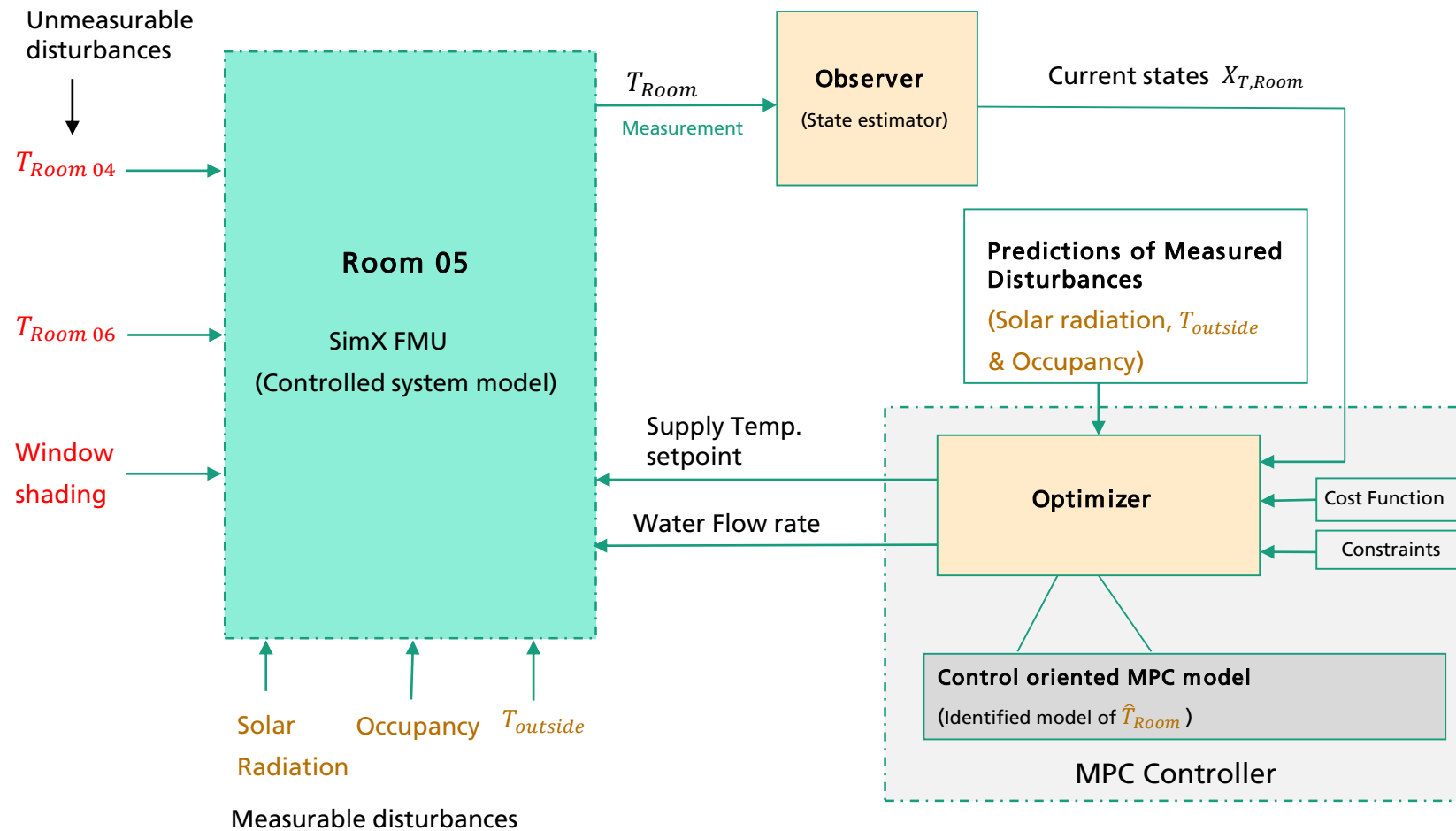
- Ziel: Regelung der Raumtemperatur eines Büros (FASA) durch Vorgabe der Vorlauftemperatur-Sollwert und Volumenstrom der Fußbodenheizung



Büro 05 Modell in SimulationX

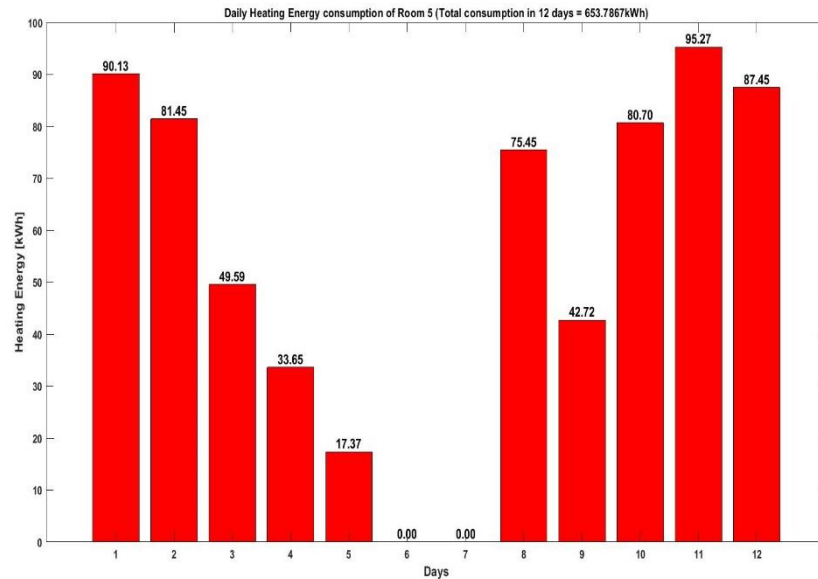
Beispiel 3: Einzelraumregelung

Regelungsstrategie



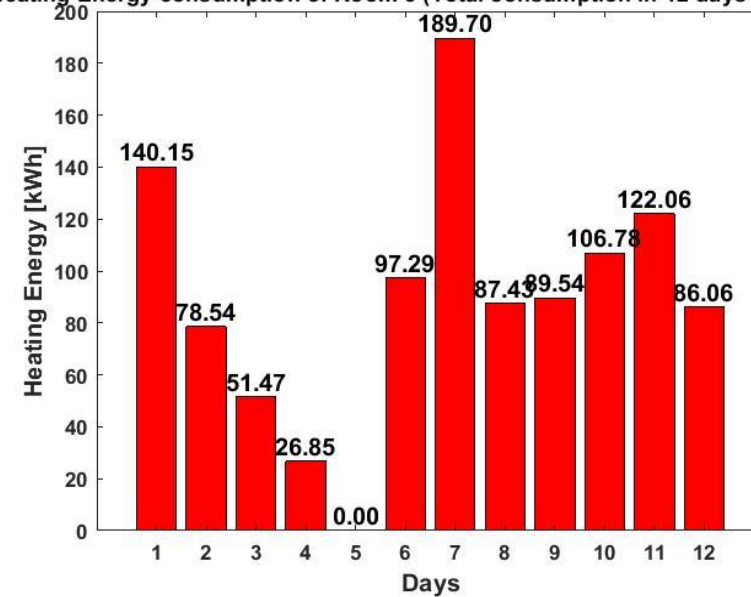
Beispiel 3: Einzelraumregelung

Ergebnisse



MPC: 654kWh

Heating Energy consumption of Room 5 (Total consumption in 12 days = 1075.81)



2-point control: 1075 kWh

Ca. 35-38% Heizenergie Einsparungen

- Anhand der Anwendungsbeispiele:
 - MPC kann erfolgreich eingesetzt werden
 - Verbunden mit Modellierungs- und Validierungsaufwand
 - Komplexität des Optimierungsproblems sollte beherrschbar sein
 - In allen Fällen konnte durch Systemidentifikation lineare Modelle approximiert werden und in die Optimierung eingesetzt werden.

- Vergleich zu Reinforcement Learning (Machine Learning)
 - RL und MPC brauchen beide gute Modelle
 - RL: Offline-Lernen möglich bzw. Lösen von komplexer Optimierungsaufgabe zu jeder Abtastzeit nicht nötig
 - RL: Gutes Lernverhalten und das Beherrschen aller Systemzustände erfordert viel Aufwand
 - MPC: Anpassung an Systemveränderungen nicht einfach möglich

Contact

Chenzi Huang & Stephan Seidel
Advanced Control Systems
Phone +49 351 45691-382 / -385
Fax +49 351 45691-111
chenzi.huang@eas.iis.fraunhofer.de
stephan.seidel@eas.iis.fraunhofer.de

Fraunhofer Institute for Integrated Circuits IIS
Division Engineering of Adaptive Systems EAS
Muenchner Strasse 16
01187 Dresden, Germany
www.eas.iis.fraunhofer.de

Thank You for Your Attention
