

**Universität
Rostock**



Traditio et Innovatio

**Modellierung und Systemidentifikation
eines Einraum-Heizsystems
mit Reinforcement Learning
KI-Pumpensteuerung**

Dissertation

zur Erlangung des akademischen Grades
doctor rerum naturalium (Dr. rer. nat.)
der Mathematisch-Naturwissenschaftlichen Fakultät
der Universität Rostock

vorgelegt von
Willi Gerbig, geb. am 30.10.1986 in Paderborn
aus Rostock

Rostock, 18. Juli 2025

Gutachter:

Prof. Dr. Kurt Frischmuth
Universität Rostock
Institut für Mathematik
Arbeitsgruppe Numerische Mathematik

Prof. Dr. Kristin de Payrebrune
RPTU Kaiserslautern-Landau
Fakultät für Maschinenbau und Verfahrenstechnik
Lehrstuhl für Computational Physics in Engineering

Datum der Einreichung: 18. Juli 2025

Datum der Verteidigung: 27. Oktober 2025

Danksagungen

An erster Stelle danke ich meinem Betreuer Prof. Dr. Kurt Frischmuth. Kurt ist, möglicherweise ohne es wirklich selbst zu wissen, viel mehr als nur ein reiner Betreuer für mich gewesen. Denn neben seinen qualifizierten fachlichen Anregungen, die mir bei der Erstellung meiner Dissertation geholfen haben, haben seine privaten Geschichten und Anekdoten sowie die fachfremden Diskussionen mit ihm meine Sicht auf das Leben und die Welt im positiven Sinne maßgebend beeinflusst. Kurt ist ein echter Mentor und Coach gewesen, von dem man viel mehr als nur fachlich profitieren konnte, wenn man ihm aufmerksam zugehört hat.

Ebenso möchte ich allen Mitgliedern der Arbeitsgruppe Numerische Mathematik danken, mit denen ich über die Jahre hinweg zusammenarbeiten durfte. Die Atmosphäre in der Gruppe ist stets von Humor und einem Arbeitsklima, das kaum hätte entspannter sein können, geprägt gewesen – ein idealer Rahmen, um produktiv zu forschen. Mein extra Dank gilt hier Prof. Dr. Klaus Neymeyr, dem Leiter dieser Arbeitsgruppe. Seine bedingungslose Unterstützung für mich und alle anderen Arbeitsgruppenmitglieder ist für mich nie selbstverständlich gewesen, ebenso wenig wie sein Führungsstil, der stets genügend Freiraum für wissenschaftliche Kreativität ließ.

Weiterhin danke ich allen Projektpartnern des SOPHi-Projektes, das die thematische Grundlage meiner Dissertation bildete. Besonders bereichernd war die Zusammenarbeit mit Kristina Kowalski vom Lehrstuhl Strömungsmaschinen der Universität Rostock. Die vielen inspirierenden Gespräche und die lustige Zeit mit ihr haben mir sehr viel Freude bereitet.

Ein herzliches Dankeschön geht an meine gesamte Familie, deren Unterstützung ich sehr schätze. Gerade meine Mutter stand meinen Brüdern und mir immer, nicht nur während meiner Promotionszeit, so aufopferungsvoll und altruistisch zur Seite, dass ich meine Dankbarkeit dafür in ein paar Worten kaum ausdrücken kann.
Danke für alles, Mama!!!

Spezieller Dank geht auch an all jene Freunde von mir, die mich mal mit konstruktiver und mal mit humorvoller Nachdrücklichkeit daran erinnern haben, dass die Dissertation endlich fertig werden sollte.

Den größten Dank hierbei verdient Saskia Kurdewan, die mich unermüdlich versucht hat zu motivieren und die neben meinem Betreuer die gesamte Arbeit sehr kurzfristig korrekturgelesen hat. Saskias fortwährenden Support weiß ich sehr zu schätzen und es ist schön zu wissen, dass es Menschen gibt, auf die man sich immer verlassen kann.

Inhaltsverzeichnis

1	Einleitung	1
2	Bilanzgleichungen zur Beschreibung physikalischer Phänomene	4
2.1	Bilanzgleichungen der Kontinuumsmechanik	4
2.2	Kirchhoffsche Regeln	9
3	Technische Grundlagen	11
3.1	Rohre	11
3.2	Thermostatventile	14
3.3	Kreiselpumpe	15
3.4	Wasser als Medium	18
3.5	Heizkessel	18
3.6	Heizkörper	20
3.7	Wände, Fenster und Türen	21
4	Modellierung	24
4.1	Modellierungssprachen und Modellierungstools	24
4.2	Synthese	36
4.3	Modellvergleich	47
5	Systemidentifikation	50
5.1	Identifikation	50
5.2	Ergebnisse der Identifikation	53
6	Etablierte Steuerstrategien	56
6.1	Formulierung eines optimalen Steuerproblems	56
6.2	Konstruktion einer Lösung nach Bellman	57
6.3	Elementare Steuerstrategien	60
6.4	Erweiterte Steuerstrategien	62
7	KI-Steuerstrategie	70
7.1	Grundlagen des Reinforcement Learnings	71
7.2	Testphase	81
7.3	Initialisierung der Steuerphase	84
7.4	Steuerphase	90
8	Vergleich und kritische Bewertung der Steuerstrategien	93

9 Fazit und Ausblick	99
Glossar	101
Symbolverzeichnis	103
Literaturverzeichnis	122
A Abbildungen	A-140
A.1 Simuliertes Nutzerverhalten	A-140
A.2 Heizkurve	A-143
A.3 Ventilkennlinie	A-143
A.4 Dymola-Modell des Chemiegebäudes der Uni Rostock . .	A-144
A.5 Benchmark-Modell in Dymola	A-145
B Tabellen	B-146
B.1 Heizkurve	B-146
B.2 Ventilkennlinien	B-146
C Parametrierung	C-148
C.1 Physikalische Konstanten	C-148
C.2 Normbedingungen für Radiatoren und Konvektoren . . .	C-148
C.3 Parametrierung des DGL-Systems	C-148
C.4 Anfangsbedingungen des DGL-Systems	C-149
C.5 Parametrierung für die Systemidentifikation	C-149
C.6 Parametrierung für die Lösung nach Bellman	C-149
C.7 Parametrierung der elementaren Steuerstrategien	C-149
C.8 Parametrierung von Grundfos' Autoadapt	C-150
C.9 Parametrierung von KSB's Dynamic Control	C-150
C.10 Parametrierung der KI-Steuerstrategie	C-150
D Ausgewählte Quelltexte des Julia-Modells	D-152
D.1 Berechnung des Betriebspunktes der Pumpe	D-152
D.2 Berechnung der Modellparameter	D-153
D.3 DAE-System	D-159
D.4 ODE-System	D-162
D.5 Hilfsfunktion für die Initialisierung der KI-Strategie . . .	D-164

1 Einleitung

Der Klimawandel, der zumindest in der Wissenschaft unumstritten ist [113, S. 184], stellt die Gesellschaft, Politik und Industrie vor große Herausforderungen. Weil seine Folgen seit Langem nahezu überall spürbar geworden sind [102, S. XIX–XXVII], hat sich im kollektiven Bewusstsein längst verankert, wie enorm wichtig es ist, die natürlichen Ressourcen der Erde zu schonen und Energie einzusparen, wo immer es geht [117, S. 26 (Abb. 4)]. Auch Pumpenhersteller wie Lowara, KSB und Grundfos beschäftigen sich mit dieser Thematik. Sie bemühen sich zur Lösung beizutragen, indem sie die Entwicklung von besseren Steueralgorithmen für ihre Pumpen mit Blick auf deren Energieeinsparpotenzial vorantreiben.

Ein konkretes Beispiel dafür ist das Forschungsprojekt „Algorithmus zur Erkennung der Nachtabsenkung in Heizungspumpen“ von Lowara in Zusammenarbeit mit der Universität Rostock, das Ende 2013 begann und Anfang 2015 abgeschlossen wurde und an dem der Autor dieser Arbeit mitgewirkt hat. Zusätzlich zu der erfolgreich entwickelten Nachtabsenkung wurden im weiteren Verlauf dieses Projektes Ideen für selbstlernende Algorithmen zur Steuerung einer Pumpe in einem Heizsystem verfolgt. Ein weiteres Beispiel stellt das vom Pumpenhersteller KSB ins Leben gerufene Forschungsprojekt „Systemische Optimierung von Pumpenregelungen zur Steigerung der Energieeffizienz von Heizungs- und Klimaanlageanlagen in der Gebäudetechnik“ (Kurztitel SOPHi) dar [32], welches vom Bundesministerium für Wirtschaft und Energie im Rahmen des Programms „Forschung für Energieoptimiertes Bauen (EnoB)“ gefördert und im Zeitraum von Mitte 2015 bis Ende 2018 unter aktiver Beteiligung des Autors realisiert wurde. In Kooperation mit der Hochschule Kaiserslautern, der Technischen Universität Kaiserslautern und der Universität Rostock hat KSB in diesem Projekt unter anderem ein ähnliches Ziel wie Lowara verfolgt. Es wurde angestrebt, die Steuerung jener Umwälzpumpen von KSB, die in Heizsystemen zum Einsatz kommen, derart zu verbessern, dass während des Betriebes Energie eingespart wird, ohne jedoch den Komfort des Nutzers zu verschlechtern. In den Forschungsabteilungen von Pumpenherstellern fungieren Einraum-Heizsysteme als Benchmark-Modelle. Infolgedessen hat der Autor der vorliegenden Dissertation im Rahmen der oben erwähnten zwei Projekte ebenfalls ein solches System, das im Wesentlichen aus einem Raum, einem Kessel, einem Heizkörper mit Thermostatventil und einer Pumpe besteht, untersucht. Im Mittelpunkt seiner Forschung stand die Entwicklung eines neuen Verfahrens anhand des Benchmark-Problems, welches sich aus der Definiti-

on eines optimalen Steuerproblems für die Drehzahl einer Pumpe in einem Einraum-Heizsystem ergab. Die im Folgenden vorgestellten Ergebnisse beruhen grundlegend auf den Ideen und Erkenntnissen, die während der Durchführung der Projekte gewonnen wurden, beinhalten jedoch viele neue Ansätze, Ausarbeitungen, Implementierungen und Untersuchungen, die weder im erstgenannten Projekt mit Lowara noch im zweitgenannten mit KSB zeitlich umsetzbar waren.

Der Hauptteil der Arbeit beginnt mit den Kapiteln 2, 3 und 4, welche unmittelbar auf diese Einleitung folgen und welche den Leser zunächst mit dem Benchmark-Modell vertraut machen. Während im Kapitel 2 alle mathematischen und physikalischen Grundlagen präsentiert werden, ist Kapitel 3 den dazugehörigen Gleichungen der einzelnen Modellkomponenten und Kapitel 4 der eigentlichen Modellierung gewidmet. Das Benchmark-Modell wurde ursprünglich von Webersinke [134] im oben erwähnten Forschungsprojekt von Lowara mit der Entwicklungsumgebung Dymola in der Programmiersprache Modelica erstellt und später von Kowalski [75] mit den gleichen Werkzeugen im SOPHi-Projekt von KSB verbessert und weiterentwickelt. Die Wahl von Modelica in Verbindung mit Dymola als Tool ist speziell für die Modellierung von komplexen, energietechnischen Gebäudesystemen typisch, weil mit der Bibliothek „BuildingSystems“ [110] sowie mit der Bibliothek „Buildings“ [137], die auch Kowalski für ihre Untersuchungen benutzt hat, zwei umfangreiche Open-Source-Bibliotheken frei zur Verfügung stehen. Für die Simulation eines verhältnismäßig kleinen Modells, wie das des hier behandelten Benchmarks eines Einraum-Heizsystems, ist jedoch bisher ungeklärt, ob diese Wahl günstig ist. Während andere Forscher für ähnliche Problemstellungen davon ausgehen, dass mit Modelica auf elegante Weise ebenso kleinere Systeme abgebildet werden können [112, S. 580 (Abschn. III)], [114, S. 1982 (Abschn. II B)], wird diese These im Kapitel der Modellierung auf den Prüfstand gestellt. Hierzu wird ein DAE-System aufgestellt, das nach Parametrierung mit denselben Werten, die im Dymola-Modell hinterlegt wurden, eine alternative Implementierung des Benchmark-Modells verkörpert. Mithilfe der Programmiersprache Julia und der Entwicklungsumgebung Visual Studio Code lässt sich das DAE-System lösen. Nach Ausarbeitung der Gemeinsamkeiten und Unterschiede zwischen Modelica und Julia beziehungsweise Dymola und Visual Studio Code, ist genug Hintergrundwissen vorhanden, um die Ergebnisse und Rechenzeiten, die sich jeweils nach derselben Simulation durch die zwei Implementierungen ergeben, zu vergleichen und zu diskutieren.

Nachdem im Kapitel 4 empirisch festgestellt wird, dass für die gegebene Problemstellung Julia und Visual Studio Code die numerisch effizienteren Werkzeuge sind als Modelica und Dymola, beschäftigt sich Kapitel 5 mit der Frage, ob die Parameter des modellierten Heizsystems auf Basis der Informationen, die an der Pumpe in der Praxis zur Verfügung stehen, identifizierbar sind. Die Beantwortung dieser Frage ist von entscheidender Relevanz für die Erfor-

schung neuer Ansätze bezüglich der Steuerung der Pumpe. Falls alle Parameter mit hinreichender Genauigkeit auf Basis der Daten, welche der Pumpe bekannt sind, identifiziert werden können, besteht die Möglichkeit modellbasierte Ansätze für die Entwicklung einer neuen Steuerung zu verfolgen. Umgekehrt rücken modellfreie Methoden in den Vordergrund, wenn diese Bedingung nicht erfüllt ist, aber Interesse an einer neuen Steuerung besteht, die nicht nur rein theoretischer Natur sein soll. Um zu entscheiden, ob eine Systemidentifikation unter den praktisch gegebenen Voraussetzungen prinzipiell möglich ist oder nicht, wird in diesem Kapitel das zuvor bereits aufgestellte DAE-System untersucht. Dem gängigen Weg der Standardliteratur [20] folgend, ergeben sich dessen Parameter aus der Lösung eines Optimierungsproblems mit einem Mehrfachschießverfahren. Die Ergebnisse der so durchgeführten Berechnungen werden zum Ende des Kapitels vorgestellt. Es zeigt sich hier, dass die Identifikation theoretisch zwar möglich, unter praktischen Bedingungen jedoch unrealistisch ist.

Das Kapitel zur Systemidentifikation offenbart damit, dass für die Entwicklung einer neuen Steuerung modellfreie Ansätze modellbasierten vorzuziehen sind. Um diese Thematik genauer erforschen zu können, bietet es sich jedoch an, zunächst ein optimales Steuerproblem für das Benchmark-Modell mathematisch zu formulieren. Dies geschieht zu Beginn von Kapitel 6. Mit der Theorie von Bellman [16] werden danach optimale Lösungen konstruiert, die zwar keine Relevanz für die Praxis haben, jedoch als Referenzlösungen für spätere Vergleiche nützlich sind. Weiter werden in diesem Kapitel elementare Steuerstrategien dargelegt, das aktuell als beste Lösung geltende Verfahren von Grundfos diskutiert [46] sowie die neue Methode von KSB aus dem SOPHi-Projekt vorgestellt.

In dem darauf folgenden, längeren Kapitel 7 stellt der Autor schließlich seine eigene Idee zum Auffinden einer Lösung des Benchmark-Problems vor. Auf Basis der Erkenntnis aus Kapitel 5 handelt es sich hierbei um einen modellfreien Reinforcement Learning Ansatz, der im Wesentlichen aus einer Testphase und einer Steuerphase besteht. In der Testphase werden gezielt verschiedene Systemzustände erzeugt, um Daten zu sammeln, welche die Grundlage für die Steuerphase bilden. In der eigentlichen Steuerphase wiederum ermöglicht es eine Funktion \tilde{q}_* , die durch ein neuronales Netz abgebildet wird, eine passende Drehzahl für den aktuellen Systemzustand zu finden. Außerdem werden die Parameter von \tilde{q}_* fortlaufend durch Auswertung der neu auftretenden Systemzustände angepasst.

Im letzten Kapitel 8 des Hauptteils der Dissertation wird ein Vergleich der Steuerstrategien aus den zwei vorangegangenen Kapiteln durchgeführt, welcher die Stärke des neuen Reinforcement Learning Ansatzes zeigt.

2 Bilanzgleichungen zur Beschreibung physikalischer Phänomene

Dieses Kapitel widmet sich zunächst der Zusammenfassung der physikalischen und mathematischen Grundlagen, die bei der Modellierung von technischen Netzwerken Anwendung finden. Dabei werden als Erstes die Bilanzgleichungen der Kontinuumsmechanik hergeleitet und erläutert, auf denen die einzelnen Komponenten des im übernächsten Kapitels erstellten Modells aufbauen. Für die Verknüpfung der einzelnen Komponenten miteinander spielen wiederum die kirchhoffschen Gesetze eine entscheidende Rolle, denen deswegen hier ein eigener Abschnitt zukommt.

2.1 Bilanzgleichungen der Kontinuumsmechanik

Die Bilanzgleichungen der Kontinuumsmechanik, welche in diesem Unterabschnitt präsentiert werden, bilden zusammen mit den Hauptsätzen der Thermodynamik sowie den kirchhoffschen Regeln, welche im nächsten Abschnitt 2.2 behandelt werden, die physikalische Basis für die Erstellung von Modellen. Dieser und der darauf folgende Unterabschnitt beruhen auf der Darstellung von Altenbach [5, S. 169–208 (Kap. 5)], wobei hier mit der allgemeinen Struktur von Bilanzgleichungen, die sowohl in der Kontinuumsmechanik als auch der Thermodynamik auftreten, gestartet wird.

Sei dazu $\mathcal{X} = \mathcal{X}(\mathcal{A}, t)$ die aktuelle Konfiguration aller materiellen Punkte eines Körpers \mathcal{K} zur Zeit t , die aus der Ausgangskonfiguration \mathcal{A} dieser materiellen Punkte zur Zeit t_0 hervorgegangen ist, und sei diese Zuordnung außerdem stetig differenzierbar und bijektiv [5, S. 75 (Def. 3.1)]. Der Übergang von \mathcal{A} nach \mathcal{X} lässt sich auf zwei äquivalente Arten ausdrücken. Bei der sogenannten lagrangeschen oder auch materiellen Betrachtungsweise wird jedes Teilchen einzeln verfolgt und nachvollzogen, an welchen Orten es sich zu welcher Zeit aufhält, wohingegen bei der eulerschen oder auch räumlichen Betrachtungsweise für jeden Ort untersucht wird, welches materielle Teilchen sich zu welcher Zeit dort befindet [5, S. 76 f. (Abschn. 3.2.1)].

Für die Volumendichteverteilung $\tilde{\Psi}$ einer mechanischen Feldgröße \mathbf{Y} in eulerscher Darstellung ergibt die Integration über das Volumen $V = V(t)$ des Körpers gerade

$$\mathbf{Y}(t) = \int_V \tilde{\Psi}(\mathbf{x}, t) dV. \quad (1)$$

Die Bilanzgleichung für diese allgemeine Größe kann als

$$\dot{Y}(t) = \frac{d}{dt} \int_V \tilde{\Psi}(\mathbf{x}, t) dV = \int_A \mathbf{n}(\mathbf{x}, t) \cdot \tilde{\Phi}(\mathbf{x}, t) dA + \int_V \tilde{\Xi}(\mathbf{x}, t) dV \quad (2)$$

geschrieben werden. Hierbei repräsentieren $\tilde{\Phi}$ die Oberflächendichte sowie $\tilde{\Xi}$ die Volumendichte der äußeren Einwirkungen auf den betrachteten Körper und der Einheitsnormalenvektor \mathbf{n} bezieht sich auf das Flächenelement dA seiner Oberfläche $A = A(t)$. Die Änderung der Bilanzgröße Y hängt also von zwei Dingen ab. Einerseits von den Zu- oder Abflüssen von Y durch die Oberfläche von \mathcal{K} und andererseits von der äußeren Zufuhr in das Volumen sowie der Erzeugung und dem Verlust von Y innerhalb des Volumens.

Äquivalenterweise lässt sich die allgemeine Bilanzgleichung (2) auch mithilfe der Massendichte ρ formulieren:

$$\begin{aligned} \dot{Y}(t) &= \frac{d}{dt} \int_m \Psi(\mathbf{x}, t) dm = \frac{d}{dt} \int_V \Psi(\mathbf{x}, t) \rho(\mathbf{x}, t) dV \\ &= \int_A \mathbf{n}(\mathbf{x}, t) \cdot \Phi(\mathbf{x}, t) dA + \int_V \Xi(\mathbf{x}, t) \rho(\mathbf{x}, t) dV. \end{aligned} \quad (3)$$

Den Dichtefunktionen Ψ und Ξ mit $\Psi\rho = \tilde{\Psi}$ und $\Xi\rho = \tilde{\Xi}$ kommt dabei eine ganz ähnliche Bedeutung zu wie zuvor, jedoch mit dem Unterschied, dass diese sich nun auf die Masse $m = m(t)$ des Körpers anstatt auf sein Volumen beziehen, wohingegen $\Phi = \tilde{\Phi}$ in dieser Darstellung unverändert bleibt. Unter der Voraussetzung, dass Ψ eine stetig differenzierbare Funktion ist, geht (3) unter Anwendung des gaußschen Integralsatzes [38, Abschn. 1] über zu

$$\frac{d}{dt} \int_V \Psi(\mathbf{x}, t) \rho(\mathbf{x}, t) dV = \int_V (\operatorname{div} \Phi(\mathbf{x}, t) + \Xi(\mathbf{x}, t) \rho(\mathbf{x}, t)) dV.$$

Weil die letzte Identität auch für jeden Teilkörper von \mathcal{K} mit beliebig kleinem Volumen gilt, liefert der Lokalisierungssatz für Integrale [61, S. 119 (Theorem 1.11.1)] die lokale Form

$$\frac{d}{dt} (\Psi(\mathbf{x}, t) \rho(\mathbf{x}, t)) = \operatorname{div} \Phi(\mathbf{x}, t) + \Xi(\mathbf{x}, t) \rho(\mathbf{x}, t)$$

der allgemeinen Bilanzgleichung.

Als Nächstes folgt die Herleitung des Transport-Theorems, welche im Wesentlichen auf die Bestimmung der zeitlichen Ableitung von Y aus Gleichung (1) aufbaut. Zu diesem Zweck wird das Integral von der Momentankonfiguration vorübergehend auf die Referenzkonfiguration transformiert, also ein kurzzeitiger Wechsel von der eulerschen zur lagrangeschen Darstellung vorgenommen. Die Funktionaldeterminante dieser Transformation sei $\det(\mathbf{F}(\mathbf{a}, t))$, die

Geschwindigkeit $\mathbf{v} = \mathbf{v}(\mathbf{a}, t) = \dot{\mathbf{x}}(\mathbf{a}, t)$ und das Volumen der Ausgangskonfiguration $V_0 = V(t_0)$. Mithilfe der Relationen $\frac{\partial}{\partial t} \det(\mathbf{F}(\mathbf{a}, t)) = \det(\mathbf{F}(\mathbf{a}, t)) \operatorname{div} \mathbf{v}(\mathbf{a}, t)$ [5, S. 90] und $\partial \tilde{\Psi}(\mathbf{a}, t) / \partial t = d\tilde{\Psi}(\mathbf{x}, t) / dt$ lautet das erste Zwischenergebnis

$$\begin{aligned} \dot{Y}(t) &= \frac{d}{dt} \int_V \tilde{\Psi}(\mathbf{x}, t) dV = \frac{d}{dt} \int_{V_0} \tilde{\Psi}(\mathbf{a}, t) \det(\mathbf{F}(\mathbf{a}, t)) dV_0 \\ &= \int_{V_0} \left[\frac{\partial \tilde{\Psi}(\mathbf{a}, t)}{\partial t} \det(\mathbf{F}(\mathbf{a}, t)) + \tilde{\Psi}(\mathbf{a}, t) \frac{\partial}{\partial t} \det(\mathbf{F}(\mathbf{a}, t)) \right] dV_0 \\ &= \int_{V_0} \left[\frac{\partial \tilde{\Psi}(\mathbf{a}, t)}{\partial t} + \tilde{\Psi}(\mathbf{a}, t) \operatorname{div} \mathbf{v}(\mathbf{a}, t) \right] \det(\mathbf{F}(\mathbf{a}, t)) dV_0 \\ &= \int_V \left[\frac{d\tilde{\Psi}(\mathbf{x}, t)}{dt} + \tilde{\Psi}(\mathbf{x}, t) \operatorname{div} \mathbf{v}(\mathbf{x}, t) \right] dV. \end{aligned}$$

Unter Ausnutzung der Ketten- und der Produktregel lässt sich der Integrand weiter umformulieren zu

$$\dot{Y}(t) = \int_V \left[\frac{\partial \tilde{\Psi}(\mathbf{x}, t)}{\partial t} + \operatorname{div} \left[\tilde{\Psi}(\mathbf{x}, t) \otimes \mathbf{v}(\mathbf{x}, t) \right] \right] dV, \quad (4)$$

und der gaußsche Integralsatz führt dann schließlich auf die folgende Form des Transport-Theorems:

$$\dot{Y}(t) = \frac{d}{dt} \int_V \tilde{\Psi}(\mathbf{x}, t) dV = \int_V \frac{\partial \tilde{\Psi}(\mathbf{x}, t)}{\partial t} dV + \int_A \mathbf{n}(\mathbf{x}, t) \cdot \left[\tilde{\Psi}(\mathbf{x}, t) \otimes \mathbf{v}(\mathbf{x}, t) \right] dA.$$

Die oben aufgestellten allgemeinen Bilanzgleichungen ermöglichen nun die Untersuchung der für die Modellierung von dynamischen Systemen wichtigen Größen der Masse, des Impulses, des Drehimpulses und der Energie. Für die durch

$$m(t) = \int_V \rho(\mathbf{x}, t) dV$$

definierte Masse eines Körpers gilt der Massenerhaltungssatz. Dieser besagt, dass m stets konstant bleibt, solange weder ein Austausch von Masse über die Oberfläche des Körpers stattfindet ($\tilde{\Phi} \equiv 0$) noch sich im Inneren des Volumens die Masse durch Zufuhr oder Erzeugung ändert ($\tilde{\Xi} \equiv 0$). Durch Einsetzen von $Y = m$ und somit $\tilde{\Psi} = \rho$ in (2), lässt sich der Massenerhaltungssatz durch

$$\dot{m}(t) = \frac{d}{dt} \int_V \rho(\mathbf{x}, t) dV = 0 \quad (5)$$

ausdrücken. Im Kontext des Transport-Theorems (4)

$$0 = \dot{m}(t) = \int_V \left[\frac{\partial \rho(\mathbf{x}, t)}{\partial t} + \operatorname{div} [\rho(\mathbf{x}, t) \mathbf{v}(\mathbf{x}, t)] \right] dV$$

führt das wiederum auf die Kontinuitätsgleichung

$$\frac{\partial \rho(\mathbf{x}, t)}{\partial t} + \operatorname{div} [\rho(\mathbf{x}, t) \mathbf{v}(\mathbf{x}, t)] = 0, \quad (6)$$

welche folglich eine lokale Beschreibung der Massenerhaltung darstellt. Speziell für inkompressible Fluide, die sich dadurch auszeichnen, dass Änderungen des Drucks nicht zu Dichteänderungen führen, wird bei der Vernachlässigung aller thermischen Effekte von $\rho(\mathbf{x}, t) \equiv \text{const.}$ ausgegangen [8, S. 75]. Aber selbst bei Berücksichtigung von thermischen Effekten weicht die Dichte eines inkompressiblen Fluids in Abhängigkeit von der Temperatur häufig nur wenig von einem mittleren Referenzwert ρ_0 ab, sodass sich die Dichte in der Näherung von Boussinesq durch $\rho = \rho_0 + \rho'(\mathbf{x}, t)$ mit $|\rho'(\mathbf{x}, t)| \ll \rho_0$ ausdrücken lässt [23, S. 83 f. (Abschn. 3.7)]. Im letzten Fall geht die Kontinuitätsgleichung 6 zunächst über zu

$$\rho_0 \cdot \operatorname{div} \mathbf{v}(\mathbf{x}, t) + \frac{\partial \rho'(\mathbf{x}, t)}{\partial t} + \operatorname{grad} \rho'(\mathbf{x}, t) \cdot \mathbf{v}(\mathbf{x}, t) + \rho'(\mathbf{x}, t) \cdot \operatorname{div} \mathbf{v}(\mathbf{x}, t) = 0,$$

was bei Vernachlässigung aller Terme mit $\rho'(\mathbf{x}, t)$ aufgrund der Voraussetzung $|\rho'(\mathbf{x}, t)| \ll \rho_0$ jedoch genauso auf die einfachere Form

$$\operatorname{div} \mathbf{v}(\mathbf{x}, t) = 0 \quad (7)$$

führt wie die Annahme einer konstanten Dichte [8, S. 75 (Gl. 2.2.5)], [23, S. 84 (Gl. 3.17)].

Die Bilanz für den durch

$$\mathbf{p}(t) = \int_m \mathbf{v}(\mathbf{x}, t) dm = \int_V \mathbf{v}(\mathbf{x}, t) \rho(\mathbf{x}, t) dV$$

gegebenen Impuls kann mithilfe der allgemeinen Bilanzgleichung (3) aufgestellt werden. Dazu wird $\Psi = \mathbf{v}$ verwendet und zum einen Φ mit dem cauchyschen Spannungstensor $\boldsymbol{\sigma}$, welcher symmetrisch ist, und zum anderen Ξ mit der Massenkraftdichte \mathbf{k} ersetzt:

$$\begin{aligned} \dot{\mathbf{p}}(t) &= \frac{d}{dt} \int_V \mathbf{v}(\mathbf{x}, t) \rho(\mathbf{x}, t) dV \\ &= \int_A \boldsymbol{\sigma}(\mathbf{x}, t) \cdot \mathbf{n}(\mathbf{x}, t) dA + \int_V \mathbf{k}(\mathbf{x}, t) \rho(\mathbf{x}, t) dV = \mathbf{F}_{\text{res}}(t). \end{aligned}$$

Die zeitliche Änderung des Impulses entspricht somit der resultierenden Kraft F_{res} aller äußeren Kräfte, die auf den Körper wirken [111, S. 154 (Gl. 4.23)]. In ganz ähnlicher Weise lässt sich die Bilanzgleichung für den Drehimpuls

$$\mathbf{L}_{\mathcal{O}}(t) = \int_V \mathbf{x} \times \mathbf{v}(\mathbf{x}, t) \rho(\mathbf{x}, t) dV$$

bezüglich eines Punktes \mathcal{O} aufstellen, indem für die allgemeinen Größen aus Relation (3) $\Psi = \mathbf{x} \times \mathbf{v}$, $\Phi = \mathbf{x} \times \boldsymbol{\sigma}$ und $\Xi = \mathbf{x} \times \mathbf{k}$ benutzt wird:

$$\begin{aligned} \dot{\mathbf{L}}_{\mathcal{O}}(t) &= \frac{d}{dt} \int_V \mathbf{x} \times \mathbf{v}(\mathbf{x}, t) \rho(\mathbf{x}, t) dV \\ &= \int_A \mathbf{x} \times \boldsymbol{\sigma}(\mathbf{x}, t) \cdot \mathbf{n}(\mathbf{x}, t) dA + \int_V \mathbf{x} \times \mathbf{k}(\mathbf{x}, t) \rho(\mathbf{x}, t) dV = \mathbf{M}_{\text{res}}(t). \end{aligned}$$

Die zeitliche Änderung des Drehimpulses entspricht also dem Drehmoment M_{res} der resultierenden Kraft aller äußeren Kräfte bezüglich des Punktes \mathcal{O} [111, S. 154 (Gl. 4.24)].

Beim Aufstellen der Energiebilanz bietet es sich an, zunächst nur die mechanische Energie zu betrachten und im zweiten Schritt die thermischen Einflüsse zu ergänzen. Die kinetische Energie $E_{\text{kin}}(t) = \frac{1}{2} \int_V \mathbf{v}(\mathbf{x}, t) \cdot \mathbf{v}(\mathbf{x}, t) \rho(\mathbf{x}, t) dV$ des Körpers \mathcal{K} sowie seine innere Energie $U(t) = \int_V u(\mathbf{x}, t) \rho(\mathbf{x}, t) dV$, welche mithilfe der spezifischen inneren Energie u definiert wird, sind die Ausgangsgrößen bei einer rein mechanischen Sichtweise. Es gilt, dass die Änderung der Gesamtenergie $W = E_{\text{kin}} + U$ mit der Leistung P_{ext} aller von außen am Körper angreifenden Kräfte übereinstimmt. Daraus resultiert die Gleichung

$$\begin{aligned} \dot{W}(t) &= \dot{E}_{\text{kin}}(t) + \dot{U}(t) = \frac{d}{dt} \int_V \left(\frac{1}{2} \mathbf{v}(\mathbf{x}, t) \cdot \mathbf{v}(\mathbf{x}, t) + u(\mathbf{x}, t) \right) \rho(\mathbf{x}, t) dV \\ &= \int_A \mathbf{v}(\mathbf{x}, t) \cdot \boldsymbol{\sigma}(\mathbf{x}, t) \cdot \mathbf{n}(\mathbf{x}, t) dA + \int_V \mathbf{v}(\mathbf{x}, t) \cdot \mathbf{k}(\mathbf{x}, t) \rho(\mathbf{x}, t) dV = P_{\text{ext}}(t), \end{aligned} \quad (8)$$

welche sich mit $\Psi = \frac{1}{2} \mathbf{v} \cdot \mathbf{v} + u$, $\Phi = \mathbf{v} \cdot \boldsymbol{\sigma}$ und $\Xi = \mathbf{v} \cdot \mathbf{k}$ wieder aus der allgemeinen Beziehung (3) herleiten lässt. Bei vielen realen Bewegungsprozessen des Körpers \mathcal{K} spielen erfahrungsgemäß stets auch thermische Effekte eine nicht zu vernachlässigende Rolle. Neben der mechanischen Energie speichert ein Körper thermische Energie, die durch Wärmezufuhr und Wärmeverluste innerhalb seines Volumens V sowie durch Wärmeströme über seine Oberfläche A beeinflusst werden kann. Indem jedem materiellen Punkt von \mathcal{K} auch eine absolute Temperatur T zugeordnet wird, lassen sich mechanische und thermische Größen miteinander koppeln und so die Betrachtungen der Kontinuumsmechanik um thermodynamische Aspekte erweitern. Für das Volumen eines Kontinuums, das sich bewegt und dabei gleichzeitig als thermodynami-

ches Systems aufgefasst wird, liefert der Erste Hauptsatz der Thermodynamik die Energiebilanz. Diese ergänzt die mechanische Bilanz aus Formel (8) durch Einbeziehung der Wärme Q des Körpers. Zu diesem Zweck wird auf der rechten Seite der Gleichung die Leistung der Wärmezufuhr

$$P_Q(t) = - \int_A \mathbf{h}(\mathbf{x}, t) \cdot \mathbf{n}(\mathbf{x}, t) \, dA + \int_V r(\mathbf{x}, t) \rho(\mathbf{x}, t) \, dV$$

hinzugefügt. Hierbei steht \mathbf{h} für den Wärmestromvektor pro Einheitsfläche und die skalare Größe r für die inneren Wärmequellen von \mathcal{K} . Per Konvention wurde das negative Vorzeichen vor dem Flächenintegral außerdem so gewählt, dass ein in den Körper gerichteter Wärmestrom eine Energiezufuhr bedeutet [73, Abschn. 5.5]. Damit ergibt sich folgende Energiebilanz:

$$\begin{aligned} \dot{W}(t) &= \dot{E}_{\text{kin}}(t) + \dot{U}(t) = \frac{d}{dt} \int_V \left(\frac{1}{2} \mathbf{v}(\mathbf{x}, t) \cdot \mathbf{v}(\mathbf{x}, t) + u(\mathbf{x}, t) \right) \rho(\mathbf{x}, t) \, dV \\ &= \int_A \mathbf{v}(\mathbf{x}, t) \cdot \boldsymbol{\sigma}(\mathbf{x}, t) \cdot \mathbf{n}(\mathbf{x}, t) \, dA + \int_V \mathbf{v}(\mathbf{x}, t) \cdot \mathbf{k}(\mathbf{x}, t) \rho(\mathbf{x}, t) \, dV \\ &\quad - \int_A \mathbf{h}(\mathbf{x}, t) \cdot \mathbf{n}(\mathbf{x}, t) \, dA + \int_V r(\mathbf{x}, t) \rho(\mathbf{x}, t) \, dV \\ &= P_{\text{ext}}(t) + P_Q(t). \end{aligned}$$

2.2 Kirchhoffsche Regeln

Ein Pumpenkreislauf setzt sich neben einer oder mehreren Pumpen aus Rohrleitungen sowie dazwischen verbauten Elementen, wie etwa Ventilen, Blenden, Strömungsvereinigungen oder Strömungsteilungen, zusammen [28]. Wenn die Pumpen in einem solchen Kreislauf einen bestimmten Volumenstrom erzeugen sollen, müssen die hydraulischen Widerstände solcher Bestandteile überwunden werden.

Zwischen hydraulischen und elektrischen Größen besteht eine Analogie, wenn die elektrische Ladung eines Stromkreises mit dem transportierten Volumenteil eines Pumpenkreislaufes in Beziehung gesetzt wird [48, S. 159]. Die elektrische Stromstärke korrespondiert dann mit dem Volumenstrom \dot{V} und die elektrische Spannung mit der Druckdifferenz Δp [48, S. 160]. Der elektrische Widerstand hingegen entspricht bei turbulent durchströmten, rauen Rohren, Thermostatventilen und diversen Formstücken dem hydraulischen Widerstand $R = \frac{\Delta p}{\dot{V}^2}$ [44, S. 38 f. u. S. 42 f.]. Obwohl die letzte Formel zeigt, dass die Analogie in Bezug auf das ohmsche Gesetz an Grenzen stößt, ist dies für die weiteren Betrachtungen in diesem Abschnitt unerheblich.

Auch wenn sie anfangs für elektrische Netzwerke formuliert wurden [69, S. 513 f.], sind die zwei kirchhoffschen Regeln, die auch als Knotensatz und

Maschensatz bezeichnet werden, aufgrund der Analogie genauso in hydraulischen Netzwerken anwendbar. Eng anlehnd an die Ausführungen von Haggmann [49, S. 24–28], geht es im Folgenden um die Formulierung der kirchhoffschen Regeln, allerdings für die entsprechenden hydraulischen Größen. In Pumpenkreisläufen gilt es zunächst zu berücksichtigen, dass den auftretenden Volumenströmen sowie den Druckdifferenzen eine Richtung zugeordnet wird. Zu diesem Zweck dient jeweils eine willkürlich vorgegebene Richtung, welche Bezugssinn genannt wird. Es bietet sich dabei jedoch an, für jedes eingebaute Element den gleichen Bezugssinn für den Volumenstrom, der durch das Element fließt, und den Druckverlust, der an dem Element entsteht, zu wählen, sodass auch hier dieser Konvention gefolgt wird. Falls der Bezugssinn und die tatsächliche Richtung übereinstimmen, so wird \dot{V} bzw. Δp mit einem positiven Vorzeichen versehen, sind sie entgegengesetzt, mit einem negativen. Weiter heißen jene Punkte, bei denen es zu Verzweigungen kommt, Knoten und die in sich geschlossenen Wege werden als Maschen bezeichnet. Der erste kirchhoffsche Satz drückt aus, dass an jedem Knoten die Summe aller zufließenden Volumenströme gleich der Summe aller abfließenden Volumenströme sein muss, weil im Knoten selbst kein Volumenteil verweilen kann. Falls die Bezugssinne für alle N Volumenströme \dot{V}_i derart gewählt werden, dass sie jeweils auf den Knoten weisen, ergibt sich die Beziehung

$$\sum_{i=1}^N \dot{V}_i = 0. \quad (9)$$

Das zweite kirchhoffsche Gesetz besagt, dass die Summe aller N Druckdifferenzen Δp_i , die in einer beliebigen Masche vorkommen, verschwindet, falls der Bezugssinn für jedes Δp_i so gesetzt wird, dass diese Richtungen entlang eines Umlaufs um die Masche zeigen. Als Formel ausgedrückt lautet dieser Sachverhalt

$$\sum_{i=1}^N \Delta p_i = 0.$$

3 Technische Grundlagen

Detailliert betrachtet werden in diesem Kapitel die wesentlichen Komponenten eines Heizsystems, zu denen Rohre, Thermostatventile, die Kreislumpumpe, Wasser als Medium, der Heizkessel, der Heizkörper sowie Wände mit und ohne Fenster und Türen gehören. Bei der Vorstellung dieser Komponenten liegt der technische Schwerpunkt hierbei jeweils auf den wichtigsten Gleichungen, von denen im weiteren Verlauf bei der Erstellung von Teilmodellen im nächsten Kapitel Gebrauch gemacht wird.

3.1 Rohre

Rohrleitungen treten in den verschiedensten Anwendungsbereichen auf, wobei zu den typischen Beispielen Kanalisationsanlagen zur Ableitung von Abwasser, Versorgungsnetze zur Bereitstellung von Wasser oder Gas- und Fernwärmeleitungen zählen [54, S. 4–7]. Speziell in Heizsystemen wird mithilfe einer Pumpe das vom Heizkessel erwärmte Wasser durch Rohre hindurch zu allen Radiatoren transportiert und anschließend abgekühlt auch wieder zum Heizkessel zurückgeführt.

In dieser Arbeit geht es um Rohre mit kreisförmigen Querschnitt A_{pipe} , deren Innendurchmesser mit d_{pipe} und deren Länge mit l_{pipe} bezeichnet wird und die so gelegen sind, dass ihre Anfangs- und Endpunkte einen Höhenunterschied von $\Delta h := h_s - h_e$ besitzen. Im Falle von waagrecht verlegten Rohren folgt somit $\Delta h = 0.0$ m. Für die Modellierung entscheidend ist der Druckverlust Δp_{pipe} über das gesamte Rohr, der dann entsteht, wenn ein Medium mit Dichte ρ hindurchströmt und welcher zum Beispiel im Buch von Grollius [45, S. 22–33 (Abschn. 2.8–2.11)], auf dessen Ausarbeitungen sich dieser Abschnitt bezieht, genauer erläutert wird. Um die entsprechende Formel für inkompressible, stationäre und isotherme Strömungen mit einer mittleren Strömungsgeschwindigkeit v herzuleiten, wird die Druckform

$$\frac{\rho \cdot v^2}{2} + \rho \cdot g \cdot h + p = \text{const.} \quad (10)$$

der Bernoulli-Gleichung als Ausgangspunkt genommen. Ihre Gültigkeit beschränkt sich auf ideale Flüssigkeiten, bei denen keine inneren Reibungsverluste auftreten. Die äquivalente Energieform, die sich durch Division durch ρ ergibt, ist ein Ausdruck der Erhaltung der Gesamtenergie des durch das Rohr

strömenden Mediums entlang des Stromfadens. Beim ersten Summanden dieser Bilanz handelt es sich dabei um die kinetische, beim zweiten um die potentielle und beim dritten um die Druckenergie. Wenn ein Rohr allerdings von einer realen Flüssigkeit durchströmt wird, bleiben Reibungseffekte nicht aus, sodass ein Teil der Gesamtenergie in Wärmeenergie umgewandelt wird. Die Verluste durch die Reibung wirken sich aber nur auf die Druckenergie aus. Das liegt zum einen daran, dass die Höhenlagen h_s und h_e am Anfang und Ende des Rohres fest sind und zum anderen daran, dass aus der Kontinuitätsgleichung (7) für inkompressible Flüssigkeiten unter der Annahme einer konstanten Querschnittsfläche A_{pipe} gerade $v_s = v_e = v$ für die entsprechenden Strömungsgeschwindigkeiten resultiert. Mit den Drücken p_s und p_e , die am Anfang und Ende des Rohres herrschen, sowie mit dem durch die Reibung verursachten Verlust Δp_f lässt sich auf Basis von Formel (10) folglich zunächst die Identität

$$\frac{\rho \cdot v_s^2}{2} + \rho \cdot g \cdot h_s + p_s = \frac{\rho \cdot v_e^2}{2} + \rho \cdot g \cdot h_e + p_e + \Delta p_f$$

gewinnen, die mit der Bezeichnung $\Delta p_{\text{pipe}} := p_s - p_e$ dann zu

$$\Delta p_{\text{pipe}} = \Delta p_f + \rho \cdot g \cdot \Delta h \quad (11)$$

umgeformt werden kann. Der erste Summand dieser Formel lässt sich mithilfe der Rohrreibungszahl λ_f gemäß der Darcy-Weisbach-Gleichung [101, S. 138 (Gl. 7.4)] ganz allgemein ausdrücken durch

$$\Delta p_f = \lambda_f \cdot \frac{l_{\text{pipe}}}{d_{\text{pipe}}} \cdot \frac{\rho \cdot v^2}{2}. \quad (12)$$

Die Berechnung der Rohrreibungszahl selbst hängt von der Art der Strömung ab, die wiederum durch die Reynolds-Zahl

$$Re = \frac{\rho \cdot v \cdot d_{\text{pipe}}}{\eta} \quad (13)$$

charakterisiert wird, welche mit der dynamischen Viskosität η das dimensionslose Verhältnis zwischen den Trägheitskräften und den Zähigkeitskräften der Strömung angibt [62, S. 54]. Es existieren laminare und turbulente Strömungsformen. Die ersteren sind einerseits dadurch gekennzeichnet, dass das Wasser das Rohr in geordneten, voneinander getrennten Schichten durchläuft, und andererseits dadurch, dass ihre Stromlinien mit der Rohrachse parallel sind. Bei den letzteren hingegen gilt dies nicht mehr, weil die Wasserteilchen sich beliebig in Längs- und Querrichtung bewegen, sodass überall Wirbel auftreten, die die Strömung durchmischen. Für laminare Strömungen erfolgt die Ermittlung von λ_f unter den oben beschriebenen Voraussetzungen meist durch

die explizite Relation

$$\lambda_f = \frac{64}{Re} \quad (14)$$

und für turbulente Strömungen wird die implizite Colebrook-White-Gleichung

$$\frac{1}{\sqrt{\lambda_f}} = -2 \log \left(\frac{k_{\text{pipe}} / d_{\text{pipe}}}{3.71} + \frac{2.51}{Re \sqrt{\lambda_f}} \right) \quad (15)$$

genutzt, in die auch die absolute Wandraugigkeit k_{pipe} miteingeht. Eine näherungsweise Lösung von (15) nach λ_f , die wegen ihrer Einfachheit oft genutzt wird, lautet

$$\lambda_f = \frac{0.25}{\left(\log \left(\frac{k_{\text{pipe}} / d_{\text{pipe}}}{3.71} + \frac{5.74}{Re^{0.9}} \right) \right)^2} \quad (16)$$

[125, S. 662 (Gl. 19)]. Obwohl in der Literatur auch eine kritische Reynolds-Zahl mit einem Wert von ungefähr $Re_{\text{krit}} = 2320$ angegeben wird, bei der Strömungen in geraden Rohren mit kreisförmigem Querschnitt von laminarem zu turbulenten Verhalten wechseln, gibt es viel mehr einen Bereich $Re_1 \leq Re \leq Re_2$, in dem eine laminare Strömung mit zunehmender Reynolds-Zahl allmählich „instabiler“ wird, bis sie schließlich gänzlich turbulent ist. Für diese beiden Grenzen wird zum Beispiel

$$Re_1 = 745 \cdot \exp \left(\frac{0.0065}{k_{\text{pipe}} / d_{\text{pipe}}} \right) \quad \text{und} \quad Re_2 = 4000$$

gewählt [56, S. 17]. Bei der Modellierung erfolgt die Berechnung der Rohrreibungszahl λ_f für $Re < Re_1$ dann gemäß der Formel (14), für $Re > Re_2$ gemäß der Formel (16) und im instabilen Übergangsbereich mithilfe eines kubischen Polynoms, das diese beiden Funktionen stetig differenzierbar miteinander verbindet [96, Z. 2884–2920]. Folglich geht der Druckverlust aus Gleichung (11) für den laminaren Bereich mit $Re < Re_1$ durch Einsetzen von (14) und unter Berücksichtigung von (12) sowie (13) über zu

$$\Delta p_{\text{pipe},l} = 32 \cdot \frac{\eta \cdot l_{\text{pipe}}}{d_{\text{pipe}}^2} \cdot v + \rho \cdot g \cdot \Delta h, \quad (17)$$

und für den turbulenten Bereich mit $Re > Re_2$ durch Einsetzen von (16) zu

$$\Delta p_{\text{pipe},t} = \frac{0.25}{\left(\log \left(\frac{k_{\text{pipe}} / d_{\text{pipe}}}{3.71} + 5.74 \cdot (\eta / (\rho \cdot v \cdot d_{\text{pipe}}))^{0.9} \right) \right)^2} \cdot \frac{l_{\text{pipe}}}{d_{\text{pipe}}} \cdot \frac{\rho \cdot v^2}{2} + \rho \cdot g \cdot \Delta h.$$

3.2 Thermostatventile

Schon seit der geänderten Fassung der alten Energiesparverordnung von 2013 müssen in Deutschland Eigentümer von Gebäuden, in denen Heizungsanlagen mit Wasser als Wärmeträger installiert sind, dafür sorgen, dass Einrichtungen vorhanden sind, die selbsttätig raumweise die Raumtemperatur regeln, wobei nur für Fußbodenheizungen unter bestimmten Bedingungen Ausnahmen gelten (vgl. Verordnung über energiesparenden Wärmeschutz und energiesparende Anlagentechnik bei Gebäuden in der Fassung der Bekanntmachung vom 24. Juli 2007 (BGBl. I S. 1519), die durch die Zweite Verordnung zur Änderung der Energieeinsparverordnung vom 18. November 2013 (BGBl. I S. 3951) geändert worden ist, § 14 Absatz 2). Im ersten Band des Sammelwerks von Albers [4, S. 1197–1200 (Abschn. 2.5.1-1.1)], aus dem ein Großteil der Informationen für diesen Unterabschnitt stammt, wird im Hinblick auf die Erfüllung dieser Vorschrift eine Kategorisierung zwischen elektronischen Einzelraumtemperaturregelungen und thermostatischen Heizkörperventilen vorgenommen. In der Kategorie der Thermostatventile wird dabei nochmal unterschieden zwischen Reglern, die Hilfsenergie, etwa elektrischen Strom, und solchen die keine Hilfsenergie benötigen. Die Variante ohne Hilfsenergie stellt im Sinne der gesetzlichen Verpflichtungen die kostengünstigste Lösung dar, um die Raumtemperatur zu regeln. Derartige thermostatische Heizkörperventile bestehen aus zwei Bestandteilen, einem Ventilunterteil und einem Kopf. Verschiedene Modelle dieser zwei Teile lassen sich häufig beliebig miteinander kombinieren, gerade wenn beide Teile vom selben Hersteller sind [129, S. 6].

Durch Drehen des Thermostatkopfs, der einen mechanischen Regler verkörpert, kann der Nutzer eine Wunschtemperatur T_{user} für die Raumtemperatur T_{room} vorgeben. Typischerweise lässt sich zu diesem Zweck ein Wert s auf einer Skala von 0, welche manchmal durch eine Schneeflocke gekennzeichnet ist, bis 5 einstellen, wobei der Wert 3 einer Wunschtemperatur von 20°C und ein ganzer Schritt auf dieser Skala einer Änderung von 4 K entspricht [82]. Die durch den Nutzer vorgegebene Solltemperatur in Abhängigkeit von der Einstellung s am Thermostatkopf genügt also der Formel

$$T_{\text{user}} = (s \cdot 4 + 8)^\circ\text{C} = T_0 + s \cdot 4\text{ K} + 8\text{ K}.$$

Hierbei steht $T_0 = 273.15\text{ K}$ für den Nullpunkt der Celsius-Temperaturskala.

Ein Ventil öffnet oder schließt sich prinzipiell in Abhängigkeit von der Temperaturdifferenz $\Delta T := T_{\text{room}} - T_{\text{user}}$, wobei sein Öffnungsgrad in dieser Arbeit α genannt wird. Das Öffnen und Schließen geschieht über eine gasförmige, flüssige oder pastenartige Substanz, die sich in einem Behälter im Thermostatkopf befindet und als Temperaturfühler fungiert. Sie dehnt sich bei Erwärmung so aus, dass ein Ventilstift gegen die Rückstellkraft einer Feder den Durchfluss im Ventil verringert, wohingegen sie sich beim Abkühlen wieder zusammen-

zieht, was dazu führt, dass der Ventilstift durch die Feder zurückgedrückt wird und sich der Durchfluss wieder erhöht [118]. Dabei ist wichtig zu verstehen, dass ein Nutzer mit der Einstellung s lediglich die Spannung der Rückstellfeder und damit im Endeffekt die Temperatur, bei der das Ventil schließt, beeinflussen kann. Aus Sicht der Regelungstechnik handelt es sich bei dieser Art von Thermostatventilen um Proportional-Regler mit einem Proportionalbereich von etwa 1 K, falls das Dehnstoffelement im Kopf ein Gas ist, und von etwa 2 K im Falle einer Flüssigkeit. Der Regler im Thermostatkopf schließt je nach Typ folglich, wenn die Raumtemperatur T_{room} die Solltemperatur T_{user} um einen Wert von 1 K oder 2 K übersteigt und steuert dadurch die von den Radiatoren an den Raum abgegebene Wärmemenge so, dass die Raumtemperatur dem Sollwert, den der Nutzer sich wünscht, möglichst nahekommt.

Demgegenüber dient das Ventilunterteil dem hydraulischen Abgleich, also dem Vorgang, bei dem zunächst die optimale Heizlast für das ganze Gebäude abgeschätzt oder, noch genauer, für jeden einzelnen Raum berechnet wird, um anschließend dafür zu sorgen, dass die in den einzelnen Strängen tatsächlich auftretenden Volumenströme den ermittelten Soll-Volumenströmen entsprechen [122, S. 3 u. S. 6]. Dies kann unter anderem dadurch erreicht werden, dass die richtige Voreinstellung am Ventilunterteil, mit der sich der Durchfluss unter Einhaltung einer hinreichend kleinen Toleranz begrenzen lässt, vorgenommen wird [129, S. 7].

3.3 Kreiselpumpe

Eine Kreiselpumpe ist eine Strömungsmaschine, mit der eine Flüssigkeit gefördert wird und die dafür bestimmt ist, einen festgelegten Volumenstrom \dot{V} gegen den Widerstand der Anlage, in der sie verbaut wurde, zu realisieren [47, S. 43]. Bei diesem Vorgang wird ein Druckniveau Δp erreicht, das von der aktuellen Drehzahl n der Pumpe abhängt. Das Betriebsverhalten wird durch Kennfelder charakterisiert, die vom Hersteller für jedes Pumpenmodell individuell an einem Prüfstand ermittelt werden [126], [71, S. 5]. Deshalb reicht im einfachsten Fall für die Modellierung einer Pumpe die Angabe des \dot{V} - H -Kennfeldes bereits aus. Dieses besteht aus Pumpenkennlinien $H_n(\dot{V})$, welche auch Drosselkurven genannt werden und für die eine feste Drehzahl n den Zusammenhang zwischen dem von der Pumpe geförderten Volumenstrom \dot{V} und der dabei von der Pumpe erreichten Förderhöhe H beschreiben [71, S. 5]. Die letztgenannte Größe H ist gemäß

$$\Delta p = \rho g H \tag{18}$$

proportional zu dem gesamten Druckverlust Δp der Anlage, wobei ρ die Dichte des geförderten Mediums und g die Schwerebeschleunigung sind [47, S. 48].

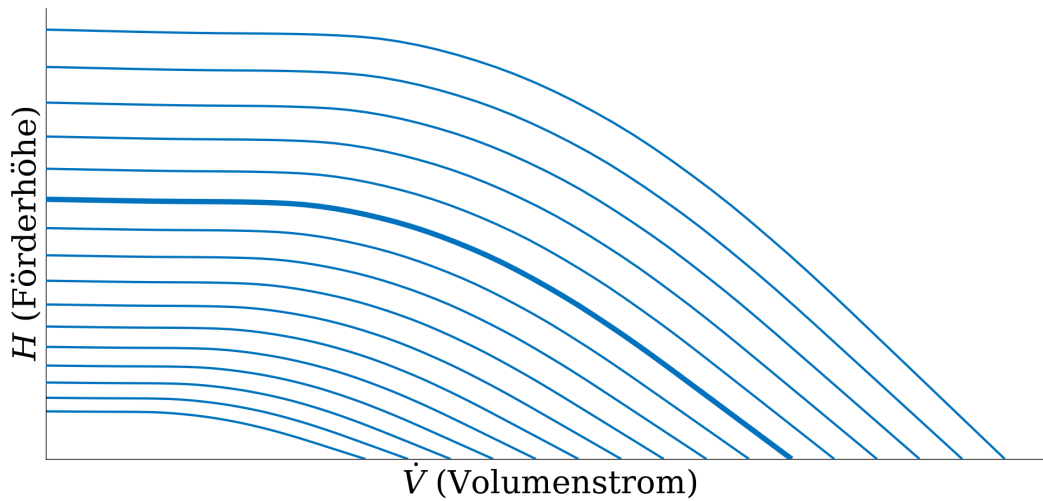


Abb. 3.1: \dot{V} - H -Kennfeld einer Pumpe, das mit den Affinitätsgesetzen auf Basis einer Referenz-Drosselkurve (dick) berechnet wurde

Die Ähnlichkeitsgesetze

$$\dot{V}_u = \dot{V}_{\text{ref}} \cdot \left(\frac{n_u}{n_{\text{ref}}} \right) \quad \text{und} \quad H_u = H_{\text{ref}} \cdot \left(\frac{n_u}{n_{\text{ref}}} \right)^2 \quad (19)$$

[136, S. 103] ermöglichen in sehr guter Näherung [50, Folie 11] das gesamte \dot{V} - H -Kennfeld zu berechnen, wenn für eine Referenzdrehzahl n_{ref} die Drosselkurve $H_{n_{\text{ref}}}(\dot{V})$ bekannt ist, weil so für weitere Drehzahlen n_u die zugehörigen Drosselkurven $H_{n_u}(\dot{V})$ ermittelt werden können. Die Abbildung 3.1 zeigt, wie das \dot{V} - H -Kennfeld einer Pumpe aussieht, wenn alle Pumpenkennlinien mithilfe der Affinitätsgesetze (19) auf Basis einer gegebenen Referenz-Drosselkurve bestimmt werden. Des Weiteren lässt sich gut erkennen, dass alle Pumpenkennlinien die Eigenschaft besitzen, streng monoton fallend zu sein.

Ebenfalls wichtig für die Beschreibung einer Pumpe ist die Anlagenkennlinie, welche auch unter der Bezeichnung Rohrleitungskennlinie bekannt ist. Diese gibt die Förderhöhe an, welche nach Formel (18) zu der resultierenden Druckdifferenz in der Anlage gehört, die bei einem bestimmten Volumenstrom überwunden werden muss [79]. Sie ist folglich abhängig von den hydraulischen Widerständen der einzelnen Verbraucher des Systems, etwa Rohre oder Ventile, und bildet somit indirekt die Volumenstrom- und Differenzdruckverteilung der Anlage ab [44, S. 11 f.]. Bei laminarer Strömung in einem Heizsystem, in dem waagrecht verlegte Rohre mit konstanter, kreisförmiger Querschnittsfläche verbaut sind, ist die Anlagenkennlinie wegen der Differenzdrücke aus den Gleichungen (17) und (41) und aufgrund der Beziehung $v = \frac{\dot{V}}{A_{\text{pipe}}} = \frac{\dot{V}}{\pi d_{\text{pipe}}^2/4}$ [4,

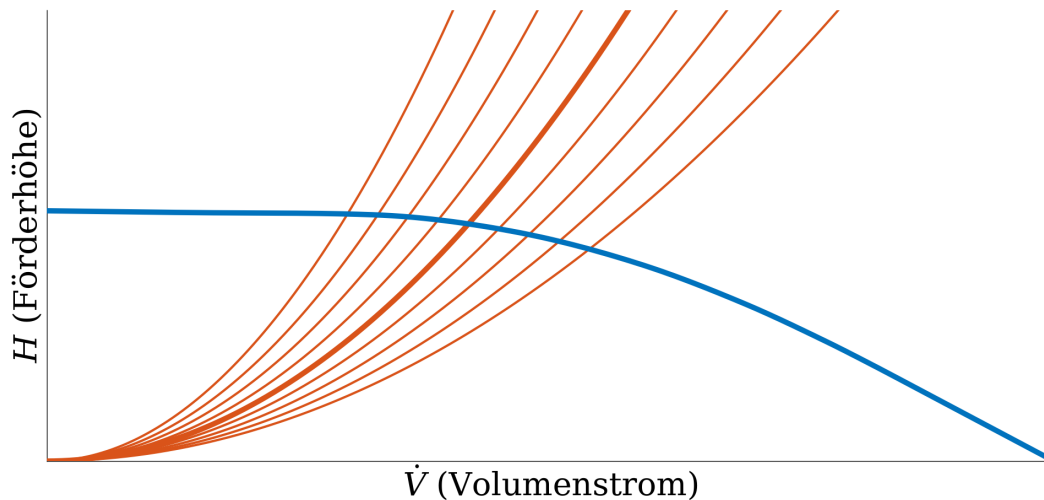


Abb. 3.2: Auswahl einer Reihe möglicher Anlagenkennlinien (orange) im \dot{V} - H -Kennfeld. Der Schnittpunkt der aktuellen Anlagenkennlinie (orange, dick) mit der aktuellen Pumpenkennlinie (blau, dick) ist der momentane Betriebspunkt der Pumpe.

S. 1279 (Abschn. 2.6.3-1.3)] beispielsweise durch

$$H_{\text{sys}} = H_{\text{valve}} + H_{\text{pipe}} \quad (20)$$

$$\Leftrightarrow H_{\text{sys}} = \frac{10^5 \text{ Pa}}{g \cdot (k_{\text{vs}} \cdot \alpha)^2 \cdot 1000 \frac{\text{kg}}{\text{m}^3}} \cdot \dot{V}^2 + 128 \cdot \frac{\eta \cdot l_{\text{pipe}}}{\pi \cdot \rho \cdot g \cdot d_{\text{pipe}}^4} \cdot \dot{V} \quad (21)$$

gegeben. Weil der Einfluss von H_{pipe} im Vergleich zu H_{valve} gering ist, kann die Rohrleitungskennlinie in guter Näherung alternativ auch als eine Parabel der Form

$$H_{\text{sys}} = a_{\text{sys}} \cdot \dot{V}^2 + H_{\text{sys},0}, \quad (22)$$

deren Scheitelpunkt auf der H -Achse liegt, beschrieben werden [79], wobei in Heizsystemen $H_{\text{sys},0} \approx 0 \text{ m}$ gilt [6, S. 32]. Die Kenntnis der aktuellen Pumpenkennlinie $H_{n(t)}(\dot{V})$ zusammen mit der Anlagenkennlinie $H_{\text{sys}}(\dot{V})$ reicht aus, um den momentanen Betriebspunkt $(\dot{V}(t), H(t))$ der Pumpe ermitteln zu können. Bei diesem handelt es sich gerade um den Schnittpunkt der beiden Kurven [4, S. 861 (Abschn. 2.2.3-4.3)]. Im \dot{V} - H -Diagramm aus Abbildung 3.2 werden exemplarisch die gegenwärtige und weitere mögliche Anlagenkennlinien eines Heizungssystems zusammen mit der aktuellen Pumpenkennlinie dargestellt, um eine Vorstellung davon zu vermitteln, wie sich der Betriebspunkt einer Pumpe ergibt.

3.4 Wasser als Medium

In einem Heizsystem ist meist Wasser das Medium, das als Wärmeträger dient und von Pumpen zu den Heizkörpern gefördert wird.

Obwohl es auch Modelle gibt, bei denen die Temperaturabhängigkeit der Stoffeigenschaften von Wasser berücksichtigt wird, reicht es häufig aus, konstante Werte zu benutzen. Das liegt daran, dass sich bei nicht allzu großen maximalen Unterschieden der auftretenden Wassertemperaturen nur kleine Veränderungen ergeben [132, S. 497 (Tab. 13.2)]. Für diesen auch hier verfolgten Ansatz werden die im Anhang C.1 aufgelisteten, konstanten Werte für die spezifischen Wärmekapazitäten c_p und c_v bei konstantem Druck und Volumen, die Dichte ρ_w , die Viskosität η_w , die Wärmeleitfähigkeit λ_w und die molare Masse M_w verwendet [98, Z. 37–68].

3.5 Heizkessel

Der Heizkessel ist ein wesentlicher Bestandteil eines jeden Heizsystems. Obwohl speziell in Neubauten hinsichtlich der Beheizungsstruktur mehr und mehr auf Elektro-Wärmepumpen (81,0 %) und kaum noch auf Strom (2,7 %) oder Fernwärme (8,2 %) gesetzt wird [15], machten diese drei Kategorien im gesamten deutschen Wohnungsraum im Jahr 2024 nur 22,4 % aus [14]. In den restlichen 77,6 % der Fälle basiert die Beheizung darauf, dass die Heizkessel die chemisch gebundene Energie des verwendeten Brennstoffs, Gas (56,1 %), Öl (17,3 %) oder Holz, Pellets und sonstige Brennstoffe (4,3 %) in thermische Energie umwandeln [52], [14]. Dies geschieht, indem durch die Verbrennung des Energieträgers zunächst die Brennkammer des Kessels erwärmt wird, wodurch sich die gewonnene Wärme anschließend auch auf das Heizwasser, welches sich in den umgebenden Rohrleitungen befindet, überträgt [127], [52].

In dem Werk von Laasch und Laasch [83, S. 676–684 (Abschn. 10.2.4)], auf das sich dieser Unterabschnitt stützt, werden nicht nur alle existierenden Kesseltypen aufgeführt, sondern auch die unterschiedlichen Möglichkeiten der Leistungsregelung, mit denen die vom Kessel gelieferte Wärmemenge an den tatsächlichen Bedarf des Heizsystems angepasst werden kann. In dieser Arbeit wird ein Verfahren modelliert, das zwar vergleichsweise einfach, aber heute gängig und recht verbreitet ist. Es beruht auf der Veränderung der Vorlauf-temperatur T_{flw} bei konstantem Volumenstrom \dot{V} statt umgekehrt. Den mathematischen Ausgangspunkt dafür bildet die Gleichung

$$\dot{Q}_{hw} = c_w \cdot \rho_w \cdot \dot{V} \cdot (T_{flw} - T_{rtn}) \quad (23)$$

[4, S. 1279] für den Wärmestrom, welcher durch das Durchströmen des vom Kessel aufgeheizten Wassers durch einen Radiator bedingt wird. Gemäß die-

ser Formel, in der T_{rtn} die Rücklauf­temperatur, $c_w = c_p = c_V$ die spezifische Wärmekapazität und ρ_w die Dichte von Wasser bezeichnen, ermöglichen es sowohl Veränderungen von T_{flw} als auch von \dot{V} die Größe \dot{Q}_{hw} zu beeinflussen. Die Steuerung der Vorlauf­temperatur erweist sich jedoch als zweckmäßiger, sodass dies die im Allgemeinen bevorzugte Methode ist. Die Heizlast, also die Wärmemenge, die vom Kessel erzeugt werden muss, um die Raumtemperatur T_{room} auf einem vorher festgelegten Niveau von beispielsweise 20°C ungefähr konstant zu halten, hängt im Wesentlichen von der Außentemperatur T_{out} ab. Wenn ein bestimmter Wert für die Raumtemperatur gewollt wird, lässt sich deswegen jeder Außentemperatur genau eine Vorlauf­temperatur T_{flw} derart zuordnen, dass die erforderliche Wärme, die zum Halten der gewünschten Temperatur im Raum erforderlich ist, vom Kessel geliefert werden kann, sobald das Wasser im Vorlauf die entsprechende Temperatur besitzt. Diesen Zusammenhang zwischen Außen- und Vorlauf­temperatur gibt die Heizkurve

$$T_{\text{flw}} = g_{\text{hc}}(T_{\text{out}}) \quad (24)$$

an. Mit Blick auf die in dieser Arbeit verwendete Kurve sei auf die Werte aus der Tabelle B.1 im Anhang B verwiesen, auf deren Grundlage eine stückweise lineare Interpolante berechnet wird. Obwohl Heizkurven normalerweise leicht nach oben gekrümmt sind [4, S. 1213], werden sie oft auch als lineare Funktionen der Form

$$T_{\text{flw}} = g_{\text{hc}}(T_{\text{out}}) := m_{\text{hc}} \cdot T_{\text{out}} + T_{\text{hc},0} \quad (25)$$

vereinfacht. Diese alternative Darstellung der Heizkurve ist als eine recht gute Näherung aufzufassen, wie aus dem Diagramm A.12 aus dem Anhang A.2 ersichtlich wird. Sie bringt zudem den Vorteil mit sich, dass sie lediglich zwei Freiheitsgerade hat. Typischerweise kann die Heizkurve vom Anwender am Kessel je nach Bedarf verstellt werden. Dabei existiert sowohl die Möglichkeit, die Steigung m_{hc} anzupassen, als auch eine die Parallelverschiebung durch Veränderung von $T_{\text{hc},0}$ vorzunehmen [4, S. 1213]. Der Außentemperaturverlauf, welcher in den später präsentierten Modellen verwendet wird, ist in Abbildung 3.3 vorzufinden. Die Daten für die Außentemperatur stammen dabei aus einem Satz von Messwerten, die im Rahmen des in der Einleitung erwähnten SOPHi-Projektes von der Gebäudeleittechnik des Chemiegebäudes der Universität Rostock im Zeitraum vom 25.11.2016 bis zum 27.02.2017 aufgezeichnet wurden. Die Kesselsteuerung, von der in dieser Arbeit ausgegangen wird, benötigt den Wert der Außentemperatur, um gemäß der vorgegebenen Heizkurve die Temperatur des Vorlaufs so zu regeln, dass diese in der Nähe des Sollwertes liegt. Ein Sensor, der normalerweise auf der sonnengeschützten und demnach ungünstigen Nordseite der Hausfassade und nicht direkt über einem Fenster angebracht wird [4, S. 1213], liefert diese Information. Die Regelung selbst basiert für Heizkessel mit festen Brennstoffen, wie sie hier nachgebildet werden

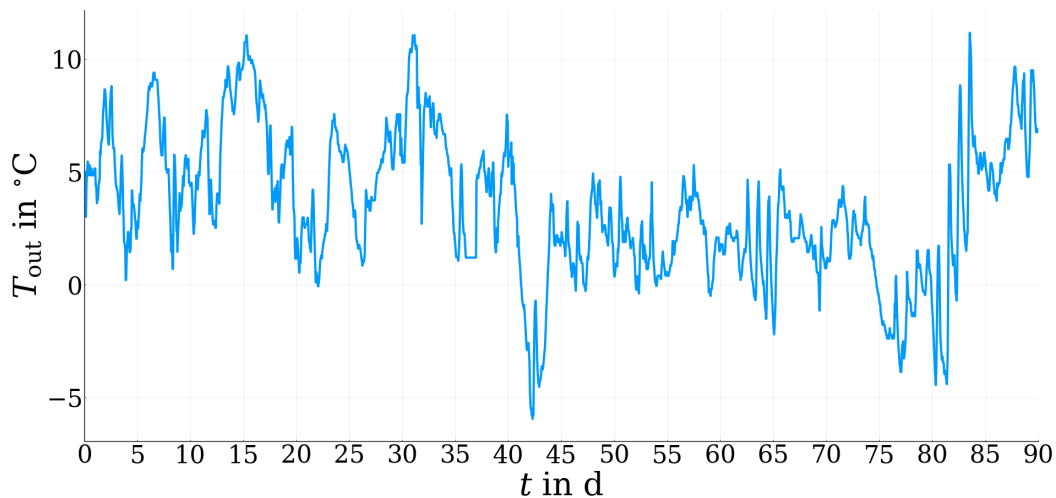


Abb. 3.3: Außentemperatur in Rostock (25.11.2016 – 27.02.2017)

sollen, auf der Änderung der Feuerleistung, indem die Luftzufuhr beeinflusst wird.

3.6 Heizkörper

Über im Raum verbaute Heizflächen kann die Wärme, die vom Heizkessel erzeugt und zunächst an das von der Pumpe geförderte Medium Wasser übertragen wird, an die Raumluft weitergereicht werden [4, S. 785 (Abschn. 2.2.2-1)]. Obwohl es Heizflächen in den verschiedensten Bauformen gibt, wie zum Beispiel Flachheizkörper, Stahl- und Gussradiatoren oder Rohr- und Rippenrohrheizkörper [4, S. 786 (Abschn. 2.2.2-1.3)], erfolgt die Wärmeabgabe an den Raum stets über zwei Wege, Konvektion auf der einen Seite und Strahlung auf der anderen [4, S. 785 f. (Abschn. 2.2.2-1.1)]. Gemäß der Darstellung im Fachmagazin von reuter onlineshop GmbH [115] wird in diesem Zusammenhang als Konvektion der Vorgang bezeichnet, bei dem kalte Luft von unten zu den Heizflächen strömt, dort erwärmt wird und dadurch zur Decke steigt, wo sie anschließend wieder abkühlt und zum Boden zurückfällt, sodass ein Kreislauf entsteht. Von den Heizflächen abgegebene Wärmestrahlung hingegen lassen sich gut mit Sonnenstrahlen vergleichen, welche auf einen Festkörper treffen [115]. Mit ihrer Energie können bei diesem Prozess die elektromagnetischen Wellen die Moleküle von Wänden, dem Rauminventar oder von im Raum befindlichen Menschen zu stärkeren Schwingungen anregen, wodurch sich die Temperatur erhöht [115].

In dieser Arbeit spielen nur glattwandig profilierte Flachheizkörper, die nach den Anforderungen der zweiteiligen Norm DIN EN 442 produziert wurden, ei-

ne Rolle. Die im zweiten Dokument vorgegebene Art der Leistungsangabe eines Heizkörpers, die auf einem genau vorgeschriebenen Prüfverfahren beruht, eignet sich hierbei gut für die Entwicklung eines Modells. Ausgehend von den Normbedingungen $T_{\text{flw,nom}}$, $T_{\text{rtn,nom}}$, $T_{\text{room,nom}}$ und $p_{\text{air,nom}}$ der Vorlauftemperatur, der Rücklauftemperatur, der Bezugs-Lufttemperatur des Raumes sowie des Luftdrucks, welche im Anhang C.2 aufgelistet sind, wird vom Hersteller für jeden Heizungstyp eine Norm-Wärmeleistung $\dot{Q}_{\text{htr,nom}}$ angegebenen [27, S. 8 f. (Abschn. 3.1.21 ff., 3.1.26, 3.1.29)]. Für den Betriebsfall lässt sich damit die eigentliche Heizleistung \dot{Q}_{htr} über die Formel

$$\begin{aligned}\dot{Q}_{\text{htr}} &= \dot{Q}_{\text{htr,nom}} \cdot \left(\frac{T_{\text{flw}} - T_{\text{rtn}}}{\ln\left(\frac{T_{\text{flw}} - T_{\text{room}}}{T_{\text{rtn}} - T_{\text{room}}}\right)} \right)^{n_{\text{htr}}} / \left(\frac{T_{\text{flw,nom}} - T_{\text{rtn,nom}}}{\ln\left(\frac{T_{\text{flw,nom}} - T_{\text{room,nom}}}{T_{\text{rtn,nom}} - T_{\text{room,nom}}}\right)} \right)^{n_{\text{htr}}} \\ &\approx \dot{Q}_{\text{htr,nom}} \cdot \left(\frac{\Delta T_{\text{ln}}}{49.8 \text{ K}} \right)^{n_{\text{htr}}}\end{aligned}\quad (26)$$

bestimmen [4, S. 787 (Abschn. 2.2.2-1.5)]. Die hier auftretende Größe $\Delta T_{\text{ln}} := (T_{\text{flw}} - T_{\text{rtn}}) / \ln\left(\frac{T_{\text{flw}} - T_{\text{room}}}{T_{\text{rtn}} - T_{\text{room}}}\right)$ wird logarithmische Übertemperatur genannt. Sie spiegelt den Unterschied zwischen der Temperatur T_{htr} des Heizkörpers und der Temperatur T_{room} der Raumluft wider [4, S. 1296]. Dazu wird eine logarithmische Mittelung für die Differenzen von Vor- und Rücklauftemperatur zur Raumtemperatur durchgeführt, wie es für Wärmeprozesse mit Temperaturgefälle üblich ist [88, Abschn. 3]. Des Weiteren heißt die dimensionslose Größe $n_{\text{htr}} \geq 1$ aus Gleichung (26) Heizkörperexponent. Dieser ist Ausdruck des im Allgemeinen nichtlinearen Zusammenhangs zwischen der Übertemperatur und der Leistungsabgabe des Heizkörpers und weist einen umso größeren Wert auf, je größer der durch die Wärmestrahlung verursachte Anteil $f_{\text{rad}} \cdot \dot{Q}_{\text{htr}}$ im Vergleich zu dem durch die Konvektion bedingten Anteil $(1 - f_{\text{rad}}) \cdot \dot{Q}_{\text{htr}}$ ist [4, S. 1296].

3.7 Wände, Fenster und Türen

Die Wände eines Gebäudes können sich aus den verschiedensten Materialien wie zum Beispiel Kalksandstein, Porenbeton, Bimsbetonstein oder Ziegelstein, aber auch aus Holzwerkstoffen wie Sperrholz oder Spanplatten sowie aus Wärmedämmstoffen wie Steinwolle, Polyurethan-Hartschaumstoff, Perlit oder Kokosfasern [78, S. 158, S. 163 u. S. 176] zusammensetzen, wobei eine Wand durchaus inhomogen sein kann, indem sie aus mehreren Schichten unterschiedlicher Baustoffe besteht. In Heizsystemen ist die Unterscheidung zwischen Innenwänden und Außenwänden, in die meist Fenster eingebettet sind, wichtig. Ebenso wie für Wände gibt es auch für die Böden und Decken von Räumen diverse Materialien.

Für eine Wand mit einer Länge l_{wall} , einer Höhe h_{wall} und einer Dicke d_{wall} werden mit Blick auf die Modellierung in dieser Arbeit drei Temperaturen definiert, die Oberflächentemperaturen $T_{\text{wall,int}}$ und $T_{\text{wall,ext}}$ auf der Innen- und Außenseite der Wand sowie die Temperatur T_{wall} in der Wand auf der Hälfte der Dicke. Bei der effektiven Wandfläche $A_{\text{wall}} = l_{\text{wall}} \cdot h_{\text{wall}} - A_{\text{win}}$ findet dabei für Außenwände gegebenenfalls die Summe der Flächen A_{win} aller in der Wand verbauten Fenster Berücksichtigung. Hinsichtlich des Wärmeaustausches mit der Umgebung spielen zwei Vorgänge eine Rolle. Zum einen kommt es innerhalb der Wand zur Wärmeleitung, welche mit den Wärmeleitfähigkeiten λ_{int} und λ_{ext} durch die Beziehungen

$$\dot{Q}_{\text{wall,int}} = \frac{\lambda_{\text{int}}}{\frac{1}{2}d_{\text{wall}}} \cdot A_{\text{wall}} \cdot (T_{\text{wall,int}} - T_{\text{wall}}) \quad (27)$$

$$\dot{Q}_{\text{wall,ext}} = \frac{\lambda_{\text{ext}}}{\frac{1}{2}d_{\text{wall}}} \cdot A_{\text{wall}} \cdot (T_{\text{wall,ext}} - T_{\text{wall}}) \quad (28)$$

für die jeweiligen Wärmeströme [4, S. 159 f. (Abschn. 1.3.5-1.1)] beschrieben werden kann. Zum anderen treten in unmittelbarer Nähe der Wandoberflächen konvektive Wärmeübergänge auf, welche mit den Wärmeübergangskoeffizienten h_{int} und h_{ext} durch die Beziehungen

$$\dot{Q}_{\text{wall,int}} = h_{\text{int}} \cdot A_{\text{wall}} \cdot (T_{\text{air}} - T_{\text{wall,int}}) \quad (29)$$

$$\dot{Q}_{\text{wall,ext}} = h_{\text{ext}} \cdot A_{\text{wall}} \cdot (T_{\text{air}} - T_{\text{wall,ext}}) \quad (30)$$

für die Wärmeströme an der Wandinnen- und Wandaußenseite [4, S. 172 f. (Abschn. 1.3.5-2)] beschrieben werden können. Die Temperatur T_{air} entspricht hierbei für die Wandinnenseite der Raumtemperatur und für die Wandaußenseite entweder der Raumtemperatur des anliegenden Nachbarräumes oder im Falle einer Außenwand der Außentemperatur. Türen, die gegebenenfalls in einer Wand verbaut sind, werden dadurch bedacht, dass die Werte der Koeffizienten λ_{int} , λ_{ext} , h_{int} und h_{ext} , die sich auf Wände ohne Türen beziehen, leicht angepasst werden. Weil die Wärmeströme aus den Formeln (27) und (29) bzw. aus (28) und (30) gleich sind, lassen sich $T_{\text{wall,int}}$ und $T_{\text{wall,ext}}$ auch eliminieren. Die Resultate nach Durchführung der Elimination lauten

$$\dot{Q}_{\text{wall,int}} = \nu_{\text{wall,int}} \cdot (T_{\text{air}} - T_{\text{wall}}) \quad (31)$$

$$\dot{Q}_{\text{wall,ext}} = \nu_{\text{wall,ext}} \cdot (T_{\text{air}} - T_{\text{wall}}) \quad (32)$$

mit dem internen und externen Wärmeleitwert

$$\nu_{\text{wall,int}} = \frac{A_{\text{wall}}}{\frac{1}{h_{\text{int}}} + \frac{1/2 \cdot d_{\text{wall}}}{\lambda_{\text{int}}}} \quad \text{und} \quad \nu_{\text{wall,ext}} = \frac{A_{\text{wall}}}{\frac{1}{h_{\text{ext}}} + \frac{1/2 \cdot d_{\text{wall}}}{\lambda_{\text{ext}}}}$$

der Wand. Böden und Decken lassen sich vom Prinzip her ähnlich behandeln wie Wände, bei denen als Umgebung statt Nachbarräumen und der Außenwelt, Räume aus dem nächsten oder vorherigen Stockwerk und den Erdboden oder das Dach Berücksichtigung finden. Die Gleichungen für die zugehörigen Wärmeströme können deswegen mit der Boden- oder Deckentemperatur T_{floor} oder T_{ceil} ganz analog zu den gerade formulierten aufgestellt werden.

Der Wärmedurchgang durch alle in einer Außenwand verbauten Fenster von der Außenluft zur Raumluft soll ebenso in ganz ähnlicher Form modelliert werden. Der Wärmedurchgangskoeffizient U_{win} liefert dafür zunächst den Wärmeleitwert $\nu_{\text{win}} = U_{\text{win}} \cdot A_{\text{win}}$, welcher wiederum auf den Wärmestrom

$$\dot{Q}_{\text{win}} = \nu_{\text{win}} \cdot (T_{\text{out}} - T_{\text{room}}) \quad (33)$$

führt.

4 Modellierung

Bei der Modellierung dynamischer Systeme gilt es stets einen Kompromiss zu finden. Einerseits sollte ein Modell so komplex sein, dass hinsichtlich des Bestimmungszwecks reale Vergleichssysteme ausreichend genau simuliert werden können, andererseits existiert jedoch oft der Wunsch, die Anzahl der Zustandsgrößen, Gleichungen und Parameter kleinzuhalten. Besonders wenn die Rechenkapazität, die zur Simulation des Modells genutzt wird, beschränkt ist oder Rechenzeiten eine entscheidende Rolle spielen, müssen während der Entwicklung auch immer die Möglichkeiten zur Reduzierung des Modells im Blick behalten werden. Aus diesem Grund soll in diesem Kapitel nicht nur auf die Modelle selbst, sondern auch auf die Tools, mit denen diese erstellt wurden, eingegangen werden. Die Tools, die in dieser Arbeit genutzt wurden, sind dabei die Software Dymola, die mit der Modellierungssprache Modelica arbeitet, sowie die relativ neue Programmiersprache Julia. Ihre jeweiligen Vor- und Nachteile werden zum Ende des Kapitels gegenübergestellt. Insbesondere wird für das Benchmark-Modell eine Implementierung in Dymola direkt mit einer rein auf mathematischer Basis erstellten Julia Implementierung verglichen.

4.1 Modellierungssprachen und Modellierungstools

4.1.1 Modelica

Modelica ist eine frei verfügbare, objektorientierte Programmiersprache, deren erste Version im September 1997, gute zwölf Monate nach dem Entwicklungsstart, veröffentlicht wurde und um deren Weiterentwicklung sich seit ihrem Gründungsjahr 2000 die Modelica Association kümmert [90, Abschn. 2]. Neben der Wartung und Verbesserung des Sprachstandards trägt die Modelica Association auch die Verantwortung für die Pflege und den Ausbau der Modelica Standard Library, einer mittlerweile mehr als eine viertel Million Zeilen Code umfassende Bibliothek, die mehr als 1300 grundlegende Modelle beinhaltet und diverse Teilbereiche der Physik abdeckt [34, S. 23], [76, S. 36 u. S. 319]. Die Sprache wurde so konzipiert, dass sie sich selbst für die Modellierung hochkomplexer Systeme eignet, die aus mehreren Komponenten der verschiedenen Teilbereiche der Physik bestehen. So kann sie etwa in der Automobilindustrie oder in der Luft- und Raumfahrt eingesetzt werden, wenn es um Anwendungen geht, bei denen Zustände des Gesamtsystems von regelbaren mechanischen, elektrischen und hydraulischen Teilsystemen abhängen [92, S. 1]. Dadurch hat

sich Modelica in den letzten Jahren als Industriestandard für die Erstellung und Simulation physikalischer Netzwerkmodelle mit konzentrierten Elementen, etabliert [21, S. IS-6]. Für die Sprache wesentlich ist außerdem, dass der Fokus bei der Erstellung von Modellen auf den dazugehörigen differentiellen, algebraischen und diskreten Gleichungen liegt [92, S. 1].

Der folgende kurze Überblick, welcher die wichtigsten Punkte des Modelica-Sprachstandards aufgreift, erleichtert es den Aufbau sowie die Simulation von Modellen, die mithilfe dieser Programmiersprache erstellt wurden, besser zu verstehen. Die Darstellung dieses Unterabschnitts beruht dabei auf der Einführung von Kral [76, Kap. 2]. Für eine ausführliche Darstellung mit vielen Erläuterungen sei auf das Buch von Fritzson [37] verwiesen sowie auf die Spezifikation für die aktuelle Modelica Version 3.6 [92], welche alle Elemente der Sprache umfasst.

Auch wenn es sich bei Modelica um eine objektorientierte Programmiersprache handelt, in der fast jedes Sprachelement eine Klasse ist [37, S. 29 (Abschn. 2.3.3)], gibt es einen fundamentalen Unterschied zu traditionellen, objektorientierten Programmiersprachen wie C++ und Java: Das auf Klassen aufbauende Programmierdesign wird in Modelica im Gegensatz zu anderen Sprachen nicht dafür genutzt, um in einfacher Weise dynamische Informationen zwischen Objekten auszutauschen, sondern es wird in erster Linie nur als ein Mittel verstanden, um physikalische Modelle möglichst übersichtlich zu strukturieren [37, S. 29 (Abschn. 2.3.3)]. Dies spiegeln auch die drei wichtigen Klassen `model`, `function` und `package` wider. Das Keyword `model` kennzeichnet Modelle für physikalische Komponenten, wobei die dazugehörigen Gleichungen mithilfe von Parametern, Variablen oder vorher definierten Konstanten und Funktionen in einer `equation`-Umgebung formuliert werden. Demgegenüber dient die `function`-Klasse dazu, mathematische Funktionen aufzustellen, welche rein algorithmisch in einer `algorithm`-Umgebung aus einer oder mehreren Eingangsvariablen eine oder mehrere Ausgangsvariablen ermitteln. Für die genaue Strukturierung von Konstanten, Funktionen und Modellen sowie allen weiteren Klassen sorgt schließlich die `package`-Klasse, mit der sinngemäß zusammengehörende Elemente zusammengefasst werden. Weil ein Package selbst auch weitere `package`-Klassen enthalten darf, besteht so die Möglichkeit eine hierarchische Ordnungsstruktur zu schaffen. Das für Objektorientierung typische Konzept der Vererbung kann in Modelica über das Schlüsselwort `extends` realisiert werden, wodurch auch Klassen selbst hierarchisch organisierbar werden und obendrein die Wiederverwertbarkeit von Programmcode gesteigert wird. Zur Wiederverwertbarkeit können auch Bestandteile einer Klassendefinition beitragen, die mit `replaceable` gekennzeichnet sind, weil sich dadurch weitere Klassen, die sich höchstens in diesen Komponenten von der Ursprungs-klasse unterscheiden, implementieren lassen, ohne den sonst übereinstimmenden Code wiederholen zu müssen [37, S. 31 ff. (Abschn. 2.5)].

Im Vergleich zu vielen anderen Programmiersprachen arbeitet Modelica le-

List. 4.1: Ausschnitt aus der Modelica-Definition des FluidPort-Konnektors [93, Z. 7–22]

```

connector FluidPort
  "Interface for quasi one-dimensional fluid flow in a piping network
  (incompressible or compressible, one or more phases, one or more
  substances)"

[...]

  flow Medium.MassFlowRate m_flow
    "Mass flow rate from the connection point into the component";
  Medium.AbsolutePressure p "Thermodynamic pressure in the connection point";

[...]

end FluidPort;

```

diglich mit vier Basis-Datentypen für Variablen: `Boolean` für Wahrheitswerte, die sich nur zu diskreten Zeitpunkten ändern können, `Integer` für ganze Zahlen, die meist als Indizes für Matrizen und Vektoren verwendet werden, `Real` für Gleitkommazahlen, die sich sogar zu physikalischen Größen mit kohärenten SI-Einheiten erweitern lassen und `String` für Zeichenketten. Zur Darstellung von komplexen Zahlen existiert daneben noch der Datentyp `Complex`, der allerdings nicht als Basis-Datentyp zählt, weil er in einer extra Bibliothek als ein `record` implementiert ist, das zwei `Real` Variablen für den Imaginär- und den Realteil der komplexen Zahl vereint. Sowohl Vektoren als auch Matrizen fallen in Modelica in die Kategorie Arrays, allerdings mit verschiedenen Dimensionen. Alle Vektoren werden als eindimensionale Arrays aufgefasst, was unter anderem dazu führt, dass nicht explizit zwischen einem Zeilen- und Spaltenvektor unterschieden werden kann [92, S. 139 (Abschn. 10.1)], und bei Matrizen handelt es sich um zweidimensionale Arrays.

Die Aufgabe, kleinere Teilmodelle zu größeren zu verbinden bis schließlich ein ganzes Simulationsmodell entstanden ist, übernimmt die `connector`-Klasse. Die auch als Ports bezeichneten Konnektoren bilden somit die Schnittstellen eines jeden Modells. Sie bestehen mit Ausnahme des speziellen Signalkonnektors mindestens aus einer Potential- und einer Flussgröße, wobei letztere durch das Schlüsselwort `flow` markiert wird. Beim `HeatPort`-Konnektor sind diese zwei Größen etwa die Temperatur T und der Wärmestrom \dot{Q} und beim `FluidPort`-Konnektor der Druck p und der Massenstrom $\dot{m} = \rho\dot{V}$. Listing 4.1 zeigt exemplarisch den entscheidenden Teil des Modelica-Codes der `FluidPort`-Klasse. Es ist wichtig zu beachten, dass zwei Modellkomponenten sich nur über zueinander kompatiblen Konnektoren miteinander verknüpfen lassen [37, S. 190. (Abschn. 5.3)], wobei es allerdings keine Einschränkung dafür gibt, mit wie

List. 4.2: Ein stark vereinfachtes Raummodell [32, S. 32 f. (Abschn. 2.1.3)], das die Funktionsweise der `connect`-Funktion veranschaulicht

```

model VerySimpleRoomModel

  import StandardRoomBib.BaseClasses.*;
  parameter Modelica.SIunits.Length height=3.42 "Raumhöhe";
  parameter Modelica.SIunits.Length width "Raumlänge";
  parameter Modelica.SIunits.Length depth "Raumtiefe";
  parameter Modelica.SIunits.CoefficientOfHeatTransfer alpha_int=8.1
    "Wärmeübergangskoeffizient Innenwand in [W/(m^2*K)]";

  replaceable RoomVolume roomvol(
    final hi=height, final wi=width, final de=depth);
  replaceable WallModel wall(
    final length1=height, final length2=width, final length3=depth,
    alpha=alpha_int, thickness = 0.075, lambda=0.1, rho=500, cp=1000);
  replaceable RadiatorModel radiator(pipeLength=width);

equation
  connect(wall.hport, roomvol.hport);
  connect(radiator.hport, roomvol.hport);

end VerySimpleRoomModel;

```

vielen anderen Konnektoren ein bestimmter Konnektor verbunden wird [37, S. 198. (Abschn. 5.4.3)]. Die Funktion `connect` übernimmt die Aufgabe Komponenten paarweise über die dazugehörigen Ports zusammenzuschalten, indem sie intern Gleichungen auf Basis der Verknüpfungen erstellt. Zu diesem Zweck werden zunächst alle erstellten Verbindungen überprüft und die Konnektoren danach so gruppiert, dass im Falle von mehrfach verknüpften Ports und den dadurch gewissermaßen entstehenden Knotenpunkten, alle Konnektoren, die zu einem solchen bestimmten Knotenpunkt gehören, in eine Gruppe fallen [92, S. 128 (Abschn. 9.2)] und ansonsten die Gruppen nur aus den zwei verbundenen Ports bestehen. Die Gleichungen, die durch diesen Prozess schließlich implizit aufgestellt werden, sehen folgendermaßen aus: Die Potentialgrößen aller Ports einer jeden Gruppe werden gleichgesetzt und die entsprechenden Flussgrößen werden unter Berücksichtigung ihres Bezugssinns gemäß dem ersten kirchhoffschen Gesetz analog zu Formel (9) aufsummiert, wodurch sich null ergibt [92, S. 128–131 (Abschn. 9.2)]. Für ein besseres Verständnis dient das konkrete Beispiel aus Listing 4.2, welches eine stark vereinfachte Version des von Kowalski entworfenen Raummodells ist. Es besteht aus lediglich drei Komponenten, dem Raumvolumen `roomvol`, der Wand `wall` und dem Heizkörper `radiator`, wobei die drei zugehörigen Teilmodelle alle einen `HeatPort`-Konnektor besitzen, über den sie durch den zweifachen Aufruf der `connect`-Funktion zu einem Knotenpunkt miteinander verknüpft werden. Die intern erstellten Gleichungen

List. 4.3: Modelica Gleichungen, welche aus den zwei Aufrufen der `connect`-Funktion aus Listing 4.2 entstehen [37, S. 198 (Abschn. 5.4.3.1)]

```
roomvol.hport.T = wall.hport.T;  
roomvol.hport.T = radiator.hport.T;  
roomvol.hport.Q_flow + wall.hport.Q_flow + radiator.hport.Q_flow = 0;
```

chungen, die daraus resultieren, beziehen sich auf die Potentialgröße der Temperatur und die Flussgröße des Wärmestroms der `HeatPort`-Klasse und sind in Listing 4.3 angegeben.

Der Umgang mit Gleichungen in Modelica ist generell auch ein sehr wesentliches Merkmal, welches diese Programmiersprache von konventionellen unterscheidet. In Modelica beruht das Erstellen von Modellen nämlich nicht direkt auf dem Prinzip von Ursache und Wirkung, sondern viel mehr auf den physikalischen Gesetzen und Zusammenhängen selbst. In diesem Kontext ist dabei die Rede von akausaler Modellierung. Das bedeutet, dass für eine in einer `equation`-Umgebung aufgestellte Gleichung a priori nicht klar ist, welche Größen der Gleichung bekannt und welche unbekannt sind. Erst nach Festlegung der Randbedingungen, mit denen das Modell arbeiten soll, kristallisiert sich dies genau heraus. Mit ein und demselben Modell ist es demnach möglich, Simulationen durchzuführen, bei denen Ergebnisse für jeweils andere unbekannte Größen berechnet werden, indem durch die Vorgabe entsprechender Randbedingungen die bekannten Größen bestimmt werden. Deswegen spielt die Reihenfolge der Gleichungen einer `equation`-Umgebung in Modelica keine Rolle und aus diesem Grund ist es ebenfalls auch unerheblich, in welcher Form eine solche Gleichung dargestellt wird. Denn das Gleichheitszeichen wird in Modelica nicht als ein Zuweisungsoperator verstanden, bei dem wie in den meisten anderen Programmiersprachen das Ergebnis der rechten Seite der Gleichung, der Variablen der linken Seite zugeordnet wird, sondern ist tatsächlich ein Ausdruck von mathematischen und physikalischen Identitäten. Die Gleichung aus Listing 4.4, bei der es sich um eine für dieses Beispiel simplifizierte Version einer Gleichung der `PartialPump`-Klasse handelt, stellt deshalb keine Berechnungsvorschrift für die Förderhöhe einer Pumpe auf Basis der Affinitätsgesetze (19) bei Kenntnis des Volumenstroms und der Drehzahl der Pumpe dar, sondern beschreibt vielmehr die mathematische Abhängigkeit dieser drei Größen. Ergo sind die drei umgestellten, alternativen Formulierungen der Gleichung, welche in Listing 4.5 auftauchen, in ihrer Wirkung äquivalent zu der ursprünglichen. Modelica bietet nichtsdestotrotz aber auch die Möglichkeit Gleichungen im programmiertechnisch klassischen Sinne zu definieren, die als Zuweisungen interpretiert werden. Die Reihenfolge solcher Anweisungen, die in einer `algorithm`-Umgebung stehen müssen und bei denen

List. 4.4: Simplifizierte Modelica Gleichung aus der PartialPump-Klasse [95, Z. 426–429]

```
head = (N/N_nominal)^2*flowCharacteristic(V_flow_single*N_nominal/N);
```

List. 4.5: Drei in Modelica äquivalente Gleichungen zu der Gleichung aus Listing 4.4

```
head/flowCharacteristic(V_flow_single*N_nominal/N) = (N/N_nominal)^2;  
flowCharacteristic(V_flow_single*N_nominal/N)/head * (N/N_nominal)^2 = 1;  
N/N_nominal = Modelica.Math.sqrt(head/  
    flowCharacteristic(V_flow_single*N_nominal/N));
```

dem Gleichheitszeichen noch ein Doppelpunkt vorausgeht, ist dann natürlich entscheidend. Weil es sich dabei um Zuordnungen handelt, darf außerdem auf der linken Seite auch nur eine Variable stehen. Derartige Algorithmen stellen eine Ausnahme dar und treten typischerweise am häufigsten bei der Definition von Funktionen auf, die dann ihrerseits beim Aufstellen von Gleichungen in einer `equation`-Umgebung Anwendung finden können.

4.1.2 Dymola

Die im Juli 1999 erschienene Version 4.0 der Software Dymola, deren Name seitdem ein Akronym für Dynamic Modeling Laboratory ist, war die erste Entwicklungsumgebung, die Modelica im vollen Umfang unterstützte [34, S. 22], [37, S. 1029]. Obwohl kostenlose Open Source Alternativen wie OpenModelica existieren, ist diese kommerzielle Lösung der Firma Dassault Systèmes AB für bestimmte Probleme die bessere Wahl, besonders wenn es um die Simulation von sehr großen Modellen mit extrem vielen Gleichungen geht [91]. Weil auch in dieser Arbeit Dymola für das Erstellen und Simulieren von Modellen genutzt wird, sollen die wichtigsten Merkmale dieses Programms kurz präsentiert werden. Zu diesem Zweck sehr nützlich ist unter anderem der erste Teil des offiziellen Benutzerhandbuchs [25, S. 13–22 (Kap. 1)], auf dessen Inhalte sich im Folgenden bezogen wird.

Die Benutzeroberfläche von Dymola besitzt zwei verschiedene Modi, den Modeling Mode, der exemplarisch in Abbildung 4.1 ausgewählt wurde und der für das Erstellen neuer oder das Bearbeiten vorhandener Modelle zur Verfügung steht, und den Simulation Mode, der die Berechnung von Simulationsergebnis-

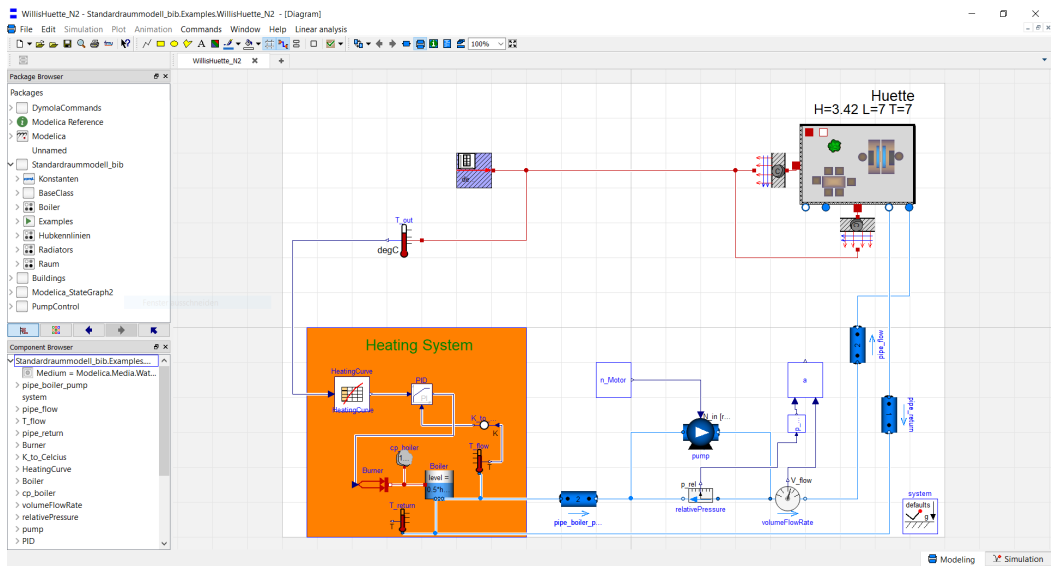


Abb. 4.1: Benutzeroberfläche des Modeling Mode von Dymola

sen und die anschließende Darstellung in Plots oder Animationen erlaubt. Im Modeling Mode wiederum kann zwischen einem rein grafischen und einem rein textbasierten Model Editor gewechselt werden. Während mit dem Text Editor direkt der Modelica-Code eines Modells oder einer seiner Komponenten angeschaut und mit Ausnahme der Standard Library bearbeitet werden kann, veranschaulicht der grafische Editor den Aufbau eines Modells bzw. Teilmodells mithilfe von Symbolen für seine einzelnen Bestandteile zusammen mit gegebenenfalls geknickten Linien für deren Verknüpfungen über die entsprechenden Konnektoren. Der im Modeling Mode in der Dymola GUI standardmäßig links unten angesiedelte Component Browser sowie der darüber stehende Package Browser erleichtern dabei die Auswahl und verbessern den Überblick. Obwohl es theoretisch möglich ist, den Modelica-Code von neuen Modellen im Text Editor von Grund auf selbst zu programmieren, bietet es sich in aller Regel zunächst an im grafischen Editor zu beginnen, indem die einzelnen benötigten Komponenten im Package Browser gesucht, ihre Symbole mit Drag and Drop in das Modellierungsfeld gezogen und sie anschließend wie gewünscht verbunden werden. Dymola generiert bei diesem Vorgang den zugehörigen Modelica-Code selbstständig [25, S. 241 (Abschn. 4.2.7, Introduction)]. Bei der darauf folgenden Parametrierung der Modelle über spezielle Programmdialoge wird der Code im Hintergrund in ähnlicher Weise ergänzt bzw. modifiziert. Nur falls anschließend die Notwendigkeit besteht, weitere Ergänzungen an gewissen Komponenten vorzunehmen, die sich nicht über die Standarddialoge realisieren lassen, etwa das Hinzufügen von Gleichungen, ist es notwendig auf den integrierten Text Editor zurückzugreifen und den bis dahin erzeugten Modelica-Quelltext

zu bearbeiten, was am speziell bei der Entwicklung eigener Bibliotheken vorkommt [25, S. 241 (Abschn. 4.2.7, Introduction)].

Nachdem ein Modell erstellt worden ist, können im Simulation Mode numerische Experimente damit durchgeführt werden. Die zwei essenziellen Bestandteile von Dymola, die dafür gebraucht werden, sind zum einen der Modelica-Compiler und zum anderen die Simulationslaufzeitumgebung [37, S. 1055 (Abschn. 19.5)]. Der Compiler wird während der Übersetzung des Modelica-Codes nicht nur prüfen, ob die vorgeschriebene Sprachsyntax eingehalten wurde, sondern insbesondere auch kontrollieren, ob alle Gleichungen so formuliert wurden, dass die physikalischen Einheiten, die den in der Gleichung auftretenden Größen bei ihrer Definition zugeordnet wurden, nicht zu einem Widerspruch führen [25, S. 396 (Abschn. 4.4.2, Introduction)]. Aus diesem Grund ist es auch empfehlenswert, alle Variablen und Parameter im Modelica-Code mit physikalischen Einheiten zu versehen, obwohl der Sprachstandard nicht dazu verpflichtet [76, S. 48 (Abschn. 2.2.3)]. Weil aufgrund des akausalen Designs von Modelica, welchem der vorherige Unterabschnitt 4.1.1 gewidmet war, Gleichungen beliebig umgestellt sein können, ist es die Aufgabe des Übersetzers, die unbekanntenen Variablen eines Modells zu identifizieren. Während dieser Vorverarbeitung werden nicht nur alle Gleichungen nach Kausalitäten sortiert, sondern auch triviale Gleichungen eliminiert sowie analytische Vereinfachungen symbolischer Natur vorgenommen [76, S. 39 (Abschn. 2.1.2)]. Am Ende des Übersetzungsprozesses steht dann zunächst ein C-Code, der bei Benutzung der Standardeinstellungen, die auf die direkte Weiterverarbeitung innerhalb von Dymola abzielen, das Modell in mathematischer Form beschreibt, etwa durch ein DAE-System, ohne jedoch ein Lösungsverfahren zu beinhalten [26, S. 363 f. (Abschn. 6.11.1)]. Die Wahl aus mehreren verfügbaren, klassischen, aber auch modernen, numerischen Solvoren zur Lösung des Systems trifft der Nutzer erst vor der Durchführung der eigentlichen Simulation [25, S. 692 f. (Abschn. 5.3.3, Simulation > Setup...)]. In dem dafür vorgesehen Programmdialog gibt es neben den Optionen, die das Verhalten des ausgewählten Solvers steuern, wie etwa der Anfangs- und Endzeit, der Schrittweite und der Toleranz [25, S. 693 (Abschn. 5.3.3, Simulation > Setup...)], auch die Möglichkeit festzulegen, welche Ergebnisvariablen am Ende der Simulation wie gespeichert werden sollen [25, S. 699 ff. (Abschn. 5.3.3, Simulation > Setup...)]. Unter Berücksichtigung der vorgenommenen Einstellungen erstellt ein C-Compiler aus dem C-Code für das Modell im nächsten Schritt eine ausführbare Programmdatei, die Dymosim genannt wird und mit deren Hilfe Dymola schließlich die Simulationsergebnisse berechnen kann [25, S. 761 (Abschn. 5.4.1, What is Dymosim?)]. Damit der Nutzer diese Resultate gut visualisieren und auswerten kann, verfügt der Simulation Mode natürlich auch über eine Reihe von Werkzeugen zum Ausgeben, Plotten und Animieren sowie über eine Exportfunktion.

Neben der Tatsache, dass Dymola problemlos und im Vergleich zu anderen Entwicklungsumgebungen auch recht effizient mit großen Modellen, die mehr als einhunderttausend Gleichungen enthalten, umgehen kann, ist eine weitere Stärke des Programms die Unterstützung des Modellexports für andere Plattformen. Die entsprechenden Programmlicenzen erlauben es zum Beispiel ein Dymola Modell für die Ausführung in Matlab/Simulink oder für Echtzeitsimulationen in Hardware in the Loop Systemen zugänglich zu machen. Dies geschieht, indem der C-Code, der während des Übersetzungsvorgangs generiert wird, in Abhängigkeit von der ausgewählten Exportvariante auf spezielle Weise ergänzt wird [26, S. 364 (Abschn. 6.11.1)].

4.1.3 Julia und Visual Studio Code

Julia ist der Name einer modernen, höheren Programmiersprache, die am MIT designt wurde und mit deren Entwicklungsstart im Jahr 2009 das Bestreben verknüpft war, ein neues Zeitalter für numerisches und wissenschaftliches Rechnen einzuläuten [63]. Als Julia im Februar 2012 als Open Source Projekt veröffentlicht wurde [18, S. 2 (Abschn. 1)], erklärten die Hauptverantwortlichen Bezanson, Karpinski, Shah und Edelman im ersten Blogeintrag auf der offiziellen Homepage <https://julialang.org/> auf leicht selbstironische Art und Weise, dass sie diese neue Sprache erschaffen hätten, weil sie gierige und unvernünftige Programmierer seien, die bei der Auswahl der Programmiersprache für eine neue Problemstellung nicht länger zwischen den jeweiligen Stärken und Schwächen abwägen wollen. Als Intensivnutzer von Matlab, als Lisp oder Perl Hacker, als Fans von Python oder Ruby, als mit Mathematica aufgewachsene Nerds oder als Meister im Plotten mit R, die sich allesamt fantastisch in C auskennen, war die Vision der Julia Entwickler eine Programmiersprache, die alle Vorteile vereint: Die Schnelligkeit von C, auf die extra großen Wert gelegt wurde, die dynamische Typisierung von Ruby, den Makroansatz von Lisp, die Notation und die Berechnungseffizienz auf dem Gebiet der linearen Algebra von Matlab, die Universalnutzbarkeit von Python, den Umgang mit Statistiken von R sowie die Verarbeitung von Strings von Perl [19].

Die wichtigsten Gründe, die Julia für viele Programmierer schon kurz nach dem Erscheinen so interessant machten [99], sollen im Folgenden erläutert werden. Dabei wird besonderes auf die Gemeinsamkeiten und Unterschiede zu anderen häufig verwendeten Programmiersprachen geachtet. Obwohl Julia zu den objektorientierten Programmiersprachen zählt [63], gibt es im Hinblick auf das Prinzip der Datenkapselung einen Unterschied zu klassischen objektorientierten Sprachen wie C++ und Java. Julia ersetzt nämlich das Paradigma, Methoden innerhalb von Klassen zu definieren und diese dadurch direkt an die jeweilige Klasse zu koppeln, durch ein neues, allgemeineres: Die Verknüpfung von Funktionen zu Klassen, üblicherweise als Typen bezeichnet, erfolgt ausschließlich über die Parameter einer Funktion [17, S. 86 (Abschn. 4.4)].

Um zu erkennen, welche Vorteile sich daraus ergeben, muss das hierarchische Typsystem inspiziert werden, an dessen Spitze sich der Datentyp `Any` befindet [128, S. 129 (Abschn. 20.2)], welcher der einzig echte statische Datentyp in Julia ist [18, S. 3 (Abschn. 2.1)]. Dieses Typsystem stellt nach den Ausführungen von Bezanson et al. [18, S. 2 (Abschn. 1)] den bestmöglichen Kompromiss zwischen dem Nutzerwunsch nach Einfachheit durch dynamische Typisierung, den Sprachen wie Python, LISP und Ruby erfüllen, und dem Performancevorteil durch statische Typisierung, den Sprachen wie C und Fortran besitzen, dar [17, S. 74 (Abschn. 3.1)]. Einerseits besteht für den gewöhnlichen Anwender an keiner Stelle die Notwendigkeit Datentypen anzugeben, weil Julia in der Lage ist, diese zur Laufzeit des Programms ohne große Einbußen in der Performance selbstständig allein abzuleiten [17, S. 68 (Abschn. 1.1) u. S. 74 (Abschn. 3.2)] und andererseits können erfahrenerer Programmierer an entscheidenden Stellen im Quellcode, etwa beim Erstellen von Bibliotheken, durch die explizite Benennung von Datentypen, den Compiler mit zusätzlichen Informationen versorgen [17, S. 78 (Abschn. 3.4)]. Des Weiteren gibt es im Gegensatz zu vielen anderen dynamischen Programmiersprachen im Grunde keinen Unterschied zwischen bereits im Sprachstandard integrierten Datentypen und solchen, die vom Nutzer neu erschaffen werden [17, S. 75 f. (Abschn. 3.3)]. Warum das eben beschriebene Typsystem so und nicht anders von den Entwicklern designt wurde, folgte aus deren grundlegender Überzeugung. Sie waren nämlich der Ansicht, dass eine Sprache für wissenschaftliches und numerisches Rechnen dann besonders schnell ist, wenn sie so konzipiert wird, dass es Sinn ergibt, all ihre Bibliotheken in der Sprache selbst zu schreiben [17, S. 67 (Abschn. 1.1)]. Julia verfolgte diesen Ansatz von Beginn an, sodass der Kern der ersten Implementierung, externe Bibliotheken nicht mitgezählt, gerade mal circa 11000 Zeilen C, 4000 Zeilen C++ und 3500 Zeilen Scheme Code umfasste, während die Standardbibliothek aus etwa 25000 Zeilen Julia Code bestand [18, S. 24 (Abschn. 5.3)].

Ein weiterer Grund, warum dieses dynamische und hierarchisch strukturierte Typsystem von so großer Bedeutung ist, offenbart sich im Zusammenhang mit der Definition von Funktionen. Bei den allermeisten Funktionen aus der Standardbibliothek handelt es sich in Julia nämlich um Multimethoden, einer Sammlung von Methoden mit demselben Namen, die sich durch die Anzahl und/oder die Datentypen ihrer Parameter voneinander unterscheiden [18, S. 5 (Abschn. 2.5)]. Multimethoden sind einer der Hauptgründe für die hohe Effizienz von Julia, weil beim Aufruf einer Funktion ein Mechanismus greift, der *multiple dispatch* genannt wird [17, S. 79 (Abschn. 4.1)]. Hierbei wird aus der entsprechenden Methodensammlung die passendste ausgewählt, indem die Anzahl und die Datentypen der beim Aufruf verwendeten Argumente mit denen der Parameter der einzelnen Methoden abgeglichen werden, welche auf Basis der Datentyp-Hierarchie so vorsortiert sind, dass der erste Datentyp-kompatible Treffer der spezifischste und damit beste ist

List. 4.6: Julia-Code für die Definition einer Funktion nach Formel (22)

```
function H_sys( V, a, H0)
    return a .* V.^2 .+ H0
end
```

[18, S. 8 (Abschn. 2.10)]. Dadurch wird erreicht, dass, falls möglich, entweder schon während der Kompilierung oder aber zur Laufzeit, immer die effizienteste Methode ausgeführt wird [17, S. 85 (Abschn. 4.3)]. Manche Programmiersprachen, wie Java, verfügen zwar mit Polymorphismus über einen auf den ersten Blick ähnlichen Mechanismus, jedoch dürfen hier die Datentypen beim Übersetzen des Programms nie unbekannt sein und die Methoden sind, wie oben schon erwähnt, direkt an eine Klasse gebunden [17, S. 85 (Fig. 4)]. Zudem machen selbst jene Programmiersprachen, die multiple dispatch ebenfalls unterstützen, wie diverse Implementationen von Common Lisp oder Dylan sowie Multi Java, mit durchschnittlich etwa 1.5 bis 2.5 Methoden pro Funktion [100, S. 573 (Fig. 11)], deutlich weniger Gebrauch davon als Julia, wo in der Standardbibliothek durchschnittlich 5.86 und für Operatoren sogar 28.13 Methoden zu einer Multimethode gehören [17, S. 87 (Tab. I)].

Als Nächstes wird ein kurzer Blick auf einige Sprachelemente von Julia geworfen und hierbei mit der Syntax gestartet. Im Gegensatz zu vielen anderen Sprachen sind eine Reihe von Unicode Symbolen für die Namensgebung von Bezeichnern zulässig, von denen griechische Buchstaben, hoch- sowie tiefgestellte Zeichen und Diakritika am häufigsten genutzt werden [86, S. 11 (Abschn. 1.4)]. Listing 4.6 enthält ein kurzes Codebeispiel dazu. Außerdem darf auf das Multiplikationszeichen zwischen einer Zahl und einer rechts davon stehenden Variablen oder Funktion verzichtet werden, einer gebräuchlichen Notation aus der Mathematik, sodass sowohl $2h$ als auch $2*h$ gültige Ausdrücke sind [86, S. 11 (Abschn. 1.4)]. Unterstriche zur Trennung von Ziffern, insbesondere, wenn sie wie bei `n_ref = 3_500` als Tausendertrennzeichen verwendet werden, können gerade bei recht großen Zahlen ebenfalls die Übersicht im Code verbessern [128, S. 40 (Abschn. 13.2)]. Neben den üblichen, primitiven Datentypen für boolesche Werte (`Bool`), Strings (`String`), ganze Zahlen (`Integer`) und Gleitkommazahlen (`Float`) [59, Abschn. 1.2] zeichnet sich Julia auch durch eine Vielfalt weiterer nützlicher Datentypen aus, für die es zahlreiche vordefinierte Funktionen gibt: `Array` für Listen, `Tuple` für geordnete Tupel, `Dict` für Wörterbücher und `Set` für ungeordnete Mengen [86, S. 18–34 (Abschn. 2.1–2.6)]. Alle Kontrollstrukturen, die für höhere Programmiersprachen Standard sind, wie etwa `if`, `else` und `elseif` für Verzweigungen und `for` sowie `while` für Schleifen, stehen natürlich ebenso zur Verfügung [128, S. 99 (Kap. 18)]. Überdies können Methoden aufgrund des `Tuple`-Typs mehrere Rückgabewerte besitzen und

Argumente werden an ihre Parameter, bei denen es sich sowohl um feste Positionsparameter als auch optionale Schlüsselwortparameter handeln kann, durch „Call-by-Sharing“ übergeben [86, S. 47–51 (Abschn. 3.4.1 f. u. 3.4.6)]. Weil viele alte, jedoch sehr effiziente und nützliche Bibliotheken existieren, die in C und Fortran geschrieben wurden, stellt Julia mit `ccall` eine Möglichkeit bereit, Funktionen aus solchen Bibliotheken ohne Umwege direkt aufzurufen [128, S. 315 (Kap. 36)], wohingegen das Einbinden von Python- bzw. R-Code zwar auch möglich ist, dafür aber erst die Installation des Paketes PyCall bzw. RCall vonnöten ist [86, S. 100 (Abschn. 7.3) u. S. 106 (Abschn. 7.4)].

Noch ein erwähnenswertes Sprachelement ist aufgrund seines großen Potenzials die Metaprogrammierung. Julia folgt in diesem Punkt dem Ansatz von Lisp, bei dem der Quellcode eines Programms durch einen Datentyp der Sprache selbst repräsentiert wird, sodass ein Programm seinen eigenen Code manipulieren kann [128, S. 217 (Kap. 27)]. Funktionen, die genau das machen, werden Makros genannt und mit dem Schlüsselwort `macro` definiert [128, S. 224 (Abschn. 27.3)]. Welcher Vorteil sich draus ergibt, wird am ehesten anhand eines konkreten Beispiels klar. Das in der Standardbibliothek definierte Makro `@horner` etwa, mit dem ein Polynom nach dem Horner-Schema ausgewertet werden kann, ist mehr als 1.4-mal schneller als eine gewöhnliche Funktion, mit der dieser Algorithmus umgesetzt wird, sodass Julia hier sogar besser ist als C [120, S. 69 ff. (Kap. 4)], da dort die Möglichkeiten der Metaprogrammierung deutlich eingeschränkter sind [128, S. 217 (Kap. 27)]. Häufig verwendete Makros, die zum Sprachstandard gehören, sind `@time`, das die Laufzeit und den verwendeten Speicher bei der Ausführung von Anweisungen ermittelt, `@code_native`, welches den finalen Assemblercode einer Anweisung nach Abschluss der Kompilierung zurückgibt und `@view`, das den Zugriff auf einzelne Teile eines Arrays gestattet, ohne eine Kopie zu erzeugen [120, S. 19 (Kap. 4), S. 46 (Kap. 3) u. S. 107 (Kap. 6)].

Um abschließend die Compiler-Performance von Julia grob einordnen zu können, wird die kleine, von den Entwicklern erdachte Vergleichsreihe ausgewertet, bei der typische Szenarien wie das Aufrufen von Funktionen, das Parsen von Strings, das Sortieren von Daten, die Benutzung von Schleifen, das Erzeugen von zufälligen Zahlen, rekursive Berechnungen und die Verwendung von Arrays in Julia und zwölf weiteren Programmiersprachen (C, LuaJIT, Rust, Go, Fortran, Java, JavaScript, Matlab, Mathematica, Python, R und Octave) auf identische Weise implementiert und unter möglichst fairen Bedingungen getestet wurden [65]. Nach Durchführung der Tests wurden alle Rechenzeiten auf die von C normiert und dann für jede Sprache das geometrische Mittel über alle Szenarien gebildet [66]. Dieses Vorgehen lieferte jeweils einen Wert für die Performance, bei dem allen Szenarien unabhängig von den absoluten Rechenzeiten die gleiche Bedeutung zukam, was relative Vergleiche untereinander ermöglichte [53, S. 51 ff.]. Hier wird ein wenig anders vorgegangen, indem die Rechenzeiten [64] nicht auf die von C, sondern auf die von Julia,

C	LuaJit	Rust	Julia	Go	Fortran	Java
0.855	0.934	0.940	1.000	1.281	1.427	2.964
JavaScript	Matlab	Mathematica	Python	R	Octave	
4.099	8.181	12.511	14.466	41.519	289.476	

Tab. 4.1: Vergleich der Performance von Julia mit zwölf anderen Sprachen

normiert werden, woraus sich die in Tabelle 4.1 gelisteten Ergebnisse ergeben. Es zeigt sich, dass Julia trotz eines dynamischen Typsystems als Viertplatziertes mit zur Spitzengruppe gehört und dass der Testsieger C im Mittel gerade mal 14.5 % besser performt, wohingegen die Performance von Fortran auf Platz sechs im Vergleich zu Julia schon 42.7 % schlechter ist. Matlab, der Konkurrent im Bereich für numerisches und wissenschaftliches Rechnen, performt sogar mehr als 8-mal schlechter und Python, der Konkurrent im Bereich universeller Anwendungen, unterliegt Julia gar um einen Faktor, der größer ist als 14.

Obwohl Modelle rein theoretisch mit einem reinen Texteditor programmiert und dann über die REPL, Julias Kommandozeilenprogramm, ausgeführt werden könnten [128, S. 27 (Kap. 11)], bietet es sich an, eine integrierte Entwicklungsumgebung (IDE) mit mehr Funktionsumfang zu nutzen. Bei der derzeit beliebtesten IDE handelt es sich um Visual Studio Code (VSCode) [24, Abschn. 4], weil das zugehörige Julia Plugin sehr umfangreich und leistungsfähig ist. Wie in Bild 4.2 zu sehen ist, integriert VSCode Julias REPL in seiner Oberfläche, führt Syntax-Highlighting für den Code durch, ermöglicht schnellen Zugriff auf die offizielle Dokumentation, zeigt alle im Workspace vorhandenen Objekte an und organisiert die von einem Programm erstellten Plots. Daneben erweisen sich diverse Buttons, zum Beispiel zum Ausführen, Anhalten und Stoppen von Programmen als nützlich. Tools zum Debuggen von Code und dem raschen Auffinden von Fehlern stehen ebenfalls zur Verfügung.

4.2 Synthese

Nachdem die wichtigsten Eigenschaften und Grundlagen von Modelica und Julia nun bekannt sind, sind alle Voraussetzungen geschaffen, um sich in diesem Abschnitt damit auseinanderzusetzen, wie mit den zugehörigen Tools, Dymola und VSCode, konkret Modelle aufgebaut werden.

4.2.1 Die physikalische Repräsentation des Benchmark-Modells in Dymola

Es sei daran erinnert, dass in Dymola vorwiegend Teilmodelle aus der Standard Library oder anderen bereits vorhandenen Bibliotheken ausgewählt und

4 Modellierung

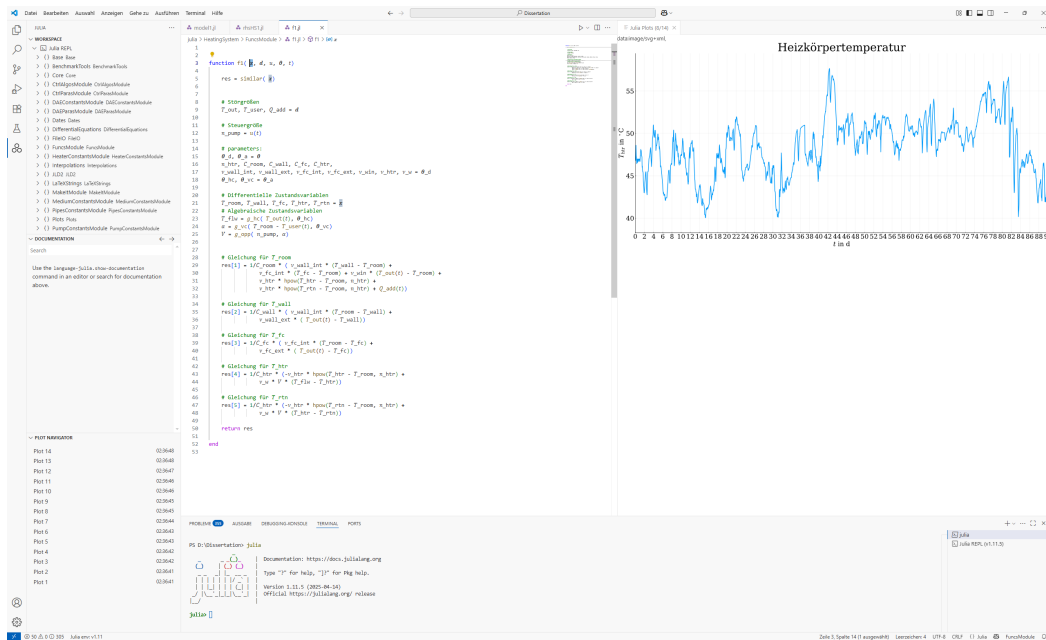


Abb. 4.2: VSCode Entwicklungsumgebung für Julia

über Konnektoren zu größeren Modellen grafisch zusammengefügt und anschließend parametrisiert werden, wodurch im Hintergrund Modelica-Code entsteht. Konnektoren, wie etwa FluidPort oder HeatPort verkörpern zusammen mit den Schnittstellen RealInput und RealOutput, den Ein- und Ausgabewerten von Funktionen, die kleinsten Bausteine von Modellen. Ihre Symbole, die in Abbildung 4.3a–4.3f dargestellt sind, finden sich deshalb bei den nächst größeren Elementen wieder. Wie in Abbildung 4.3g–4.3l zu sehen ist, enthalten beispielsweise ein CombiTable1D-Objekt, das sich etwa für die Modellierung der Außentemperatur aus dem Diagramm 3.3 durch Angabe der Messwertpaare eignet, die Summenfunktion Sum oder ein einfacher PID-Regler LimPID jeweils die RealInput-Schnittstelle 4.3e sowie die RealOutput-Schnittstelle 4.3f. Analog dazu sind die Ports 4.3c oder 4.3d Bestandteile von Objekten, die sich für die Modellierung von Wärmeprozessen eignen. Die Symbole einiger solcher Elemente, zu denen unter anderem HeatCapacitor für Wärmekapazitäten, PrescribedHeatFlow für Wärmeflüsse oder Convection für Konvektionsprozesse gehören, befinden sich in Abbildung 4.3m–4.3t. Für die wesentlichen Charakteristika eines Heizungssystems, also Kreislpumpe, Heizkessel, Rohre, Thermostatventile, Heizkörper und Wände und Fester, die im Kapitel 3 vorgestellt wurden, reichen hinsichtlich der Modellierung die Grundbausteine, die in Abbildung 4.3 gelistet sind, aus. Ein Blick auf die Diagramme der zugehörigen Modelle in Abbildung 4.4 bestätigt diese Aussage. Allerdings lässt sich nur sehr selten die Ergänzung einiger für die Funktionalität benötigter Gleichun-

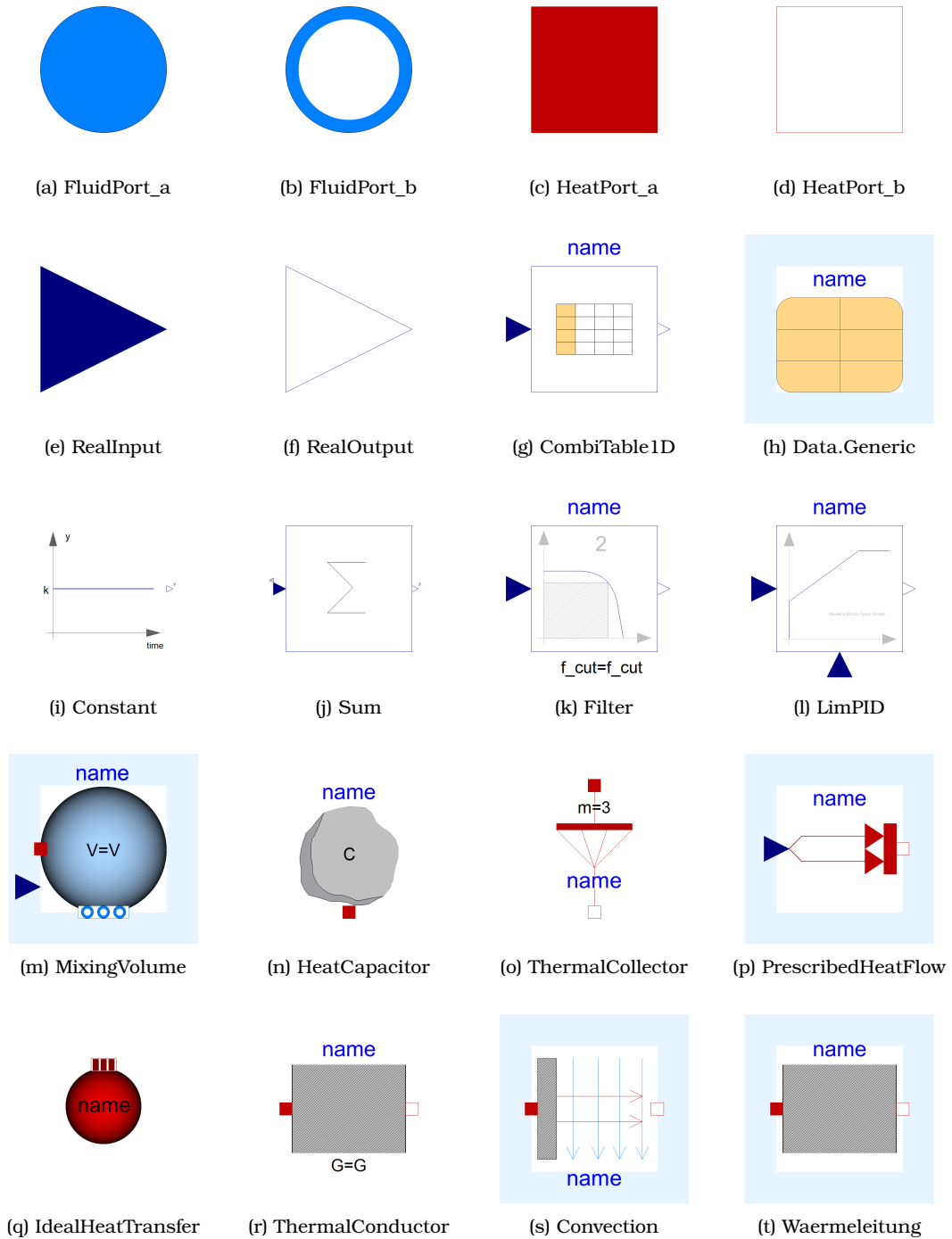


Abb. 4.3: Dymola Symbole der Grundbausteine eines Heizsystems

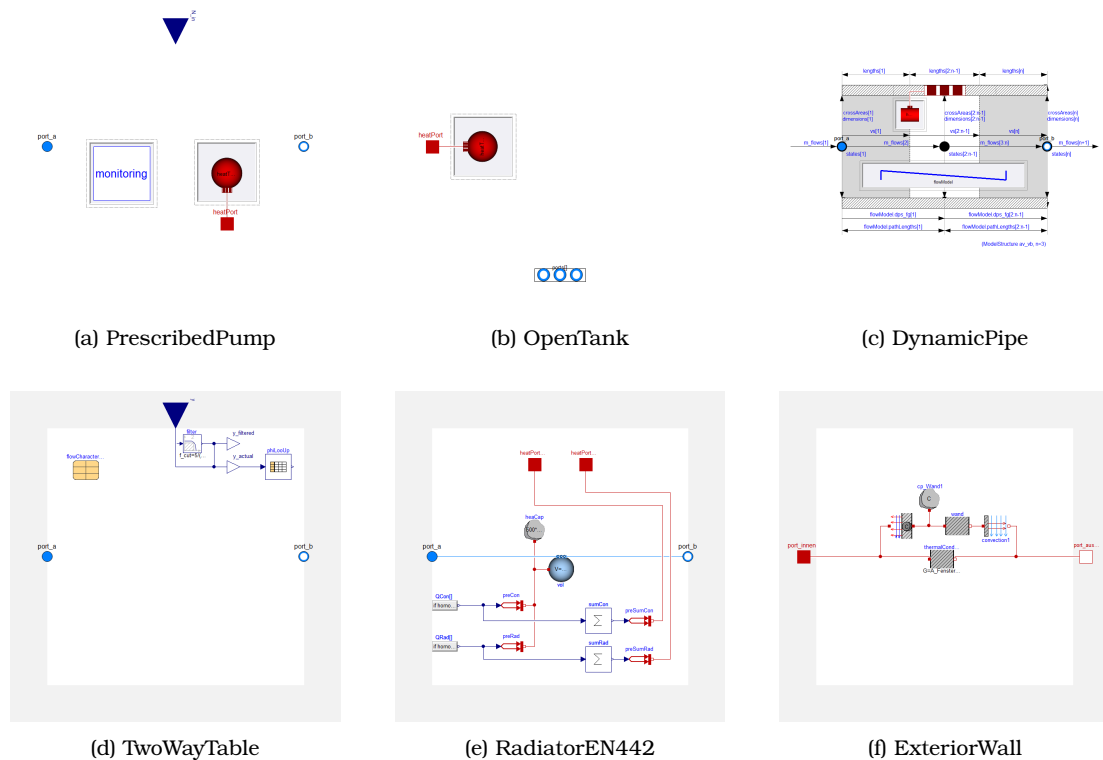


Abb. 4.4: Dymola Diagramme der Charakteristika aus Kapitel 3: Kreiselpumpe, Heizkessel, Rohre, Thermostatventil, Heizkörper und Wände

gen vermeiden. Dabei werden einerseits oft Gleichungen aufgestellt, die auf den im Abschnitt 2.1 allgemein formulierten Bilanzgleichungen beruhen und andererseits individuelle Gleichungen, auf die in den Abschnitten 3.3–3.7 eingegangen wurde. So zeigen Listing 4.7 oder 4.8 beispielsweise, wie die Bilanz für die Masse gemäß Gleichung 2 mit $Y = m$ im Modell `PartialPump` oder die mechanische Energiebilanz 8 im Modell `OpenTank` wieder auftauchen, wobei diese beiden Klassen der Modelica Standard Library angehören und Bestandteile der Implementierungen für Pumpen und Kessel sind. Die Hauptkomponenten, deren Dymola Symbole in Abbildung 4.5 zu sehen sind, lassen sich in

List. 4.7: Codezeile aus der `PartialPump`-Klasse, die auf der Bilanzgleichung 2 beruht [94, Z. 464 f.]

```
// Mass balance
mb_flow = port_a.m_flow + port_b.m_flow;
```

List. 4.8: Codeausschnitt mit Gleichungen aus der OpenTank-Klasse, die auf der mechanischen Energiebilanz aus 8 beruhen [97, Z. 87–96]

```
V = crossArea*level "Volume of fluid";
medium.p = p_ambient;

// Source termsEnergy balance
if Medium.singleState or energyDynamics == Types.Dynamics.SteadyState then
    Wb_flow = 0
    "Mechanical work is neglected, since also neglected in medium model
    (otherwise unphysical small temperature change, if tank level changes)";
else
    Wb_flow = -p_ambient*der(V);
end if;
```

Kombination mit den Grundbausteinen schließlich zu einem kompletten Heizsystem zusammenbauen. Die Abbildung A.15 im Anhang A.5 zeigt ein Modell,

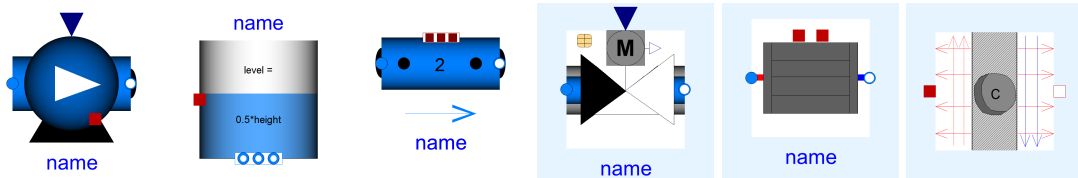


Abb. 4.5: Dymola Symbole der Charakteristika aus Kapitel 3: Kreislumppe, Heizkessel, Rohre, Thermostatventil, Heizkörper und Wände

das von Kowalski [75, Folie 5 f.] mit Unterstützung des Autors dieser Arbeit aus den oben dargestellten Komponenten auf Basis eines Ansatzes von Webersinke [134] erstellt wurde. Es handelt sich um ein Heizungssystem mit einer Pumpe, einem Kessel und einem Heizkörper, welcher einen einzelnen Raum mit Wärme versorgt. Mithilfe der hier vorgestellten Teilmodelle lassen sich aber nicht nur solche kleinen Systeme wie das Benchmark-Modell einfach zusammenfügen, sondern auch große wie etwa das eines ganzen Universitätsgebäudes. Die Abbildung A.14 im Anhang A.4 liefert am Beispiel des Chemiegebäudes der Universität Rostock den Beleg dafür [32, S. 33 f.].

4.2.2 Die Mathematik des Benchmark-Modells und seine Implementierung in Julia

Um das Dymola Benchmark-Modell aus dem vorherigen Unterabschnitt in Julia zu implementieren, muss ein differentiell-algebraisches Gleichungssystem aufgestellt werden. Bei den Zustandsgrößen, die dafür betrachtet werden, handelt es sich um die folgenden acht, von denen mit Ausnahme von T_{fc} alle bereits

im Kapitel 3 eingeführt wurden:

$$T_{\text{room}}, T_{\text{wall}}, T_{\text{fc}}, T_{\text{htr}}, T_{\text{rtn}}, T_{\text{flw}}, \alpha \text{ und } \dot{V}. \quad (34)$$

Die noch nicht definierte Größe $T_{\text{fc}} = \frac{1}{2} \cdot (T_{\text{floor}} + T_{\text{ceil}})$ sei die Durchschnittstemperatur der Bodentemperatur T_{floor} und der Deckentemperatur T_{ceil} . Diese Zusammenlegung bietet den Vorteil, die Anzahl der Gleichungen des DAE-Systems um eins reduzieren zu können. Der dadurch gleichzeitig entstehende Nachteil, T_{floor} und T_{ceil} nicht einzeln bestimmen zu können, ist für die nachfolgenden Untersuchungen hinnehmbar, weil die zwei Größen nicht von direktem Interesse sind. Die kombinierte Version aus den zur Gleichung (31) bzw. Gleichung (32) analogen Wärmeströmen $\dot{Q}_{\text{floor,int}}$ und $\dot{Q}_{\text{ceil,int}}$ bzw. $\dot{Q}_{\text{floor,ext}}$ und $\dot{Q}_{\text{ceil,ext}}$ lässt sich mithilfe von $\nu_{\text{fc,int}} := \frac{1}{2} \cdot (\nu_{\text{floor,int}} + \nu_{\text{ceil,int}})$ bzw. mithilfe von $\nu_{\text{fc,ext}} := \frac{1}{2} \cdot (\nu_{\text{floor,ext}} + \nu_{\text{ceil,ext}})$ schreiben als

$$\dot{Q}_{\text{fc,int}} = \nu_{\text{fc,int}} \cdot (T_{\text{room}} - T_{\text{fc}}) \quad \text{bzw.} \quad \dot{Q}_{\text{fc,ext}} = \nu_{\text{fc,ext}} \cdot (T_{\text{out}} - T_{\text{fc}}). \quad (35)$$

Des Weiteren wird im Modell für den Heizkörper eine Aufteilung in N gleich große Segmente mit dem gleichen Wärmeleitwert ν_{htr} vorgenommen [84, Z. 13 f.] und es werden $N + 1$ Temperaturen $T_{\text{htr},i}$ mit $T_{\text{htr},0} := T_{\text{flw}}$ und $T_{\text{htr},N} := T_{\text{rtn}}$ so festgelegt, dass dem Anfang und dem Ende jedes Segments ein $T_{\text{htr},i}$ zugeordnet ist und der Index i in Richtung der Strömung steigt. Die Wärmeabgabe des Heizkörpers an den Raum lässt sich dadurch etwas präziser ermitteln als mit Formel (26) und entspricht dann der Summe der Leistungsabgaben $\dot{Q}_{\text{htr},i}$ aller Segmente:

$$\dot{Q}_{\text{htr}} = \sum_{i=1}^N \dot{Q}_{\text{htr},i} = \sum_{i=1}^N \nu_{\text{htr}} \cdot (T_{\text{htr},i} - T_{\text{room}})^{n_{\text{htr}}}. \quad (36)$$

Ebenso splittet sich der Wärmestrom \dot{Q}_{hw} aus Relation (23) zu

$$\dot{Q}_{\text{hw}} = \sum_{i=1}^N \dot{Q}_{\text{hw},i} = \sum_{i=1}^N \nu_{\text{w}} \cdot \dot{V} \cdot (T_{\text{htr},i-1} - T_{\text{htr},i}), \quad (37)$$

wobei hier die Abkürzung $\nu_{\text{w}} := c_{\text{w}} \cdot \rho_{\text{w}}$ eingeführt wurde. Für den in dieser Arbeit behandelten speziellen Fall mit $N = 2$ steht $T_{\text{htr},1}$ für die Temperatur in der Mitte des Heizkörpers, sodass sich die Abkürzung $T_{\text{htr}} := T_{\text{htr},1}$ anbietet.

Bei den ersten fünf der in (34) aufgelisteten Zustandsgrößen handelt es sich um Temperaturen, für die mithilfe der korrespondierenden Wärmekapazitäten $C_{\text{room}}, C_{\text{wall}}, C_{\text{fc}}$ und C_{htr} die Formulierung einer differentiellen Gleichung möglich ist, indem jeweils eine Bilanz aller involvierten Wärmeströme aufgestellt wird. Mit der bisher noch nicht eingeführten Bezeichnung \dot{Q}_{add} , die alle Wärmeströme umfassen soll, welche auf zusätzliche Quellen im Raum, wie etwa menschliche Körperwärme oder die Wärme elektrischer Geräte, zurück-

zuführen sind, lauten die entsprechenden Beziehungen

$$\begin{aligned}
 C_{\text{room}} \cdot \dot{T}_{\text{room}} &= -\dot{Q}_{\text{wall,int}} - \dot{Q}_{\text{fc,int}} + \dot{Q}_{\text{win}} + \dot{Q}_{\text{htr}} + \dot{Q}_{\text{add}} \\
 C_{\text{wall}} \cdot \dot{T}_{\text{wall}} &= \dot{Q}_{\text{wall,int}} + \dot{Q}_{\text{wall,ext}} \\
 C_{\text{fc}} \cdot \dot{T}_{\text{fc}} &= \dot{Q}_{\text{fc,int}} + \dot{Q}_{\text{fc,ext}} \\
 C_{\text{htr}} \cdot \dot{T}_{\text{htr}} &= -\dot{Q}_{\text{htr,1}} + \dot{Q}_{\text{hw,1}} \\
 C_{\text{htr}} \cdot \dot{T}_{\text{rtn}} &= -\dot{Q}_{\text{htr,2}} + \dot{Q}_{\text{hw,2}}.
 \end{aligned} \tag{38}$$

Für die verbleibenden drei Zustandsgrößen T_{flw} , α und \dot{V} werden algebraische Gleichungen benutzt, welche das Verhalten des Heizkessels, des Thermostatventils und das der Pumpe beschreiben. Wie aus dem Abschnitt 3.5 ersichtlich wird, reicht für die erstgenannte Größe die Angabe der Heizkurve g_{hc} gemäß Formel (24) bereits aus.

Im Kontext des Thermostatventils spielt allerdings auch der Nutzer eine wesentliche Rolle, weil er, wie im Abschnitt 3.2 erläutert, eine Wunschtemperatur für den Raum vorgeben kann. Für die Modellierung eines realistischen Nutzerverhaltens wurden elf Profile von verschiedenen Typen von Tagesabläufen konstruiert, welche jeweils beschreiben, wann und wie ein fiktiver Nutzer im Laufe eines Tages von 0:00 Uhr bis 24:00 Uhr die Stellung s am Thermostatkopf verändert. In den Abbildungen 4.6, 4.7 und 4.8 sind davon exemplarisch drei ausgewählt und dargestellt worden. Eine vollständige Übersicht befindet sich im Anhang in den Abbildungen A.1–A.11.

Ein mehrtägiges Nutzerverhalten lässt sich mit diesen Tagesprofilen durch eine einfache Aneinanderreihung definieren. Die als Beispiel dienende Kombination

11, 3, 1, 2, 5, 9, 10, 1, 3, 3, 1, 7, 8, 9,	(1. + 2. Woche)
4, 3, 2, 2, 7, 8, 9, 1, 3, 3, 4, 5, 9, 9,	(3. + 4. Woche)
1, 3, 1, 1, 6, 9, 10, 2, 3, 3, 1, 7, 8, 9,	(5. + 6. Woche)
4, 3, 1, 2, 5, 9, 10, 1, 3, 3, 2, 5, 9, 9,	(7. + 8. Woche)
1, 3, 1, 4	(9. Woche),

die aus den Indizes von verschiedenen Tagesabläufen besteht, führt auf die in Abbildung 4.9 gezeigte Wunschtemperatur des Nutzers, welche im Folgenden auch als Solltemperatur bezeichnet wird. Eine weitere, sehr einfache, aber für Simulationszwecke sehr nützliche Kombination, die auch im späteren Verlauf der Arbeit genutzt wird, besteht ausschließlich aus dem Profil für den elften Tagesablauf und korrespondiert dadurch mit einer konstanten Solltemperatur.

Hinsichtlich der eigentlichen Modellierung des Ventils dient das Thermostatventilunterteil „V-exakt“ von Heimeier, das für typische Zweirohrsysteme mit Vor- und Rücklauf ausgelegt ist [129, S. 7], als Vorbild. Das Unterteil wird er-

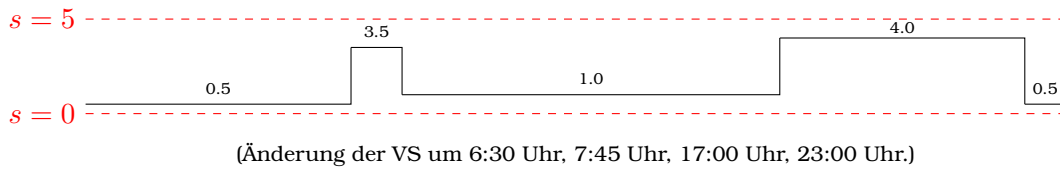


Abb. 4.6: 1. Tagesablauf: Arbeitstag, Variante 1

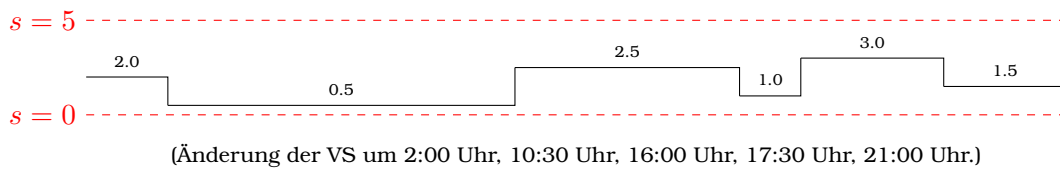


Abb. 4.7: 8. Tagesablauf: Wochenendtag, Variante 1

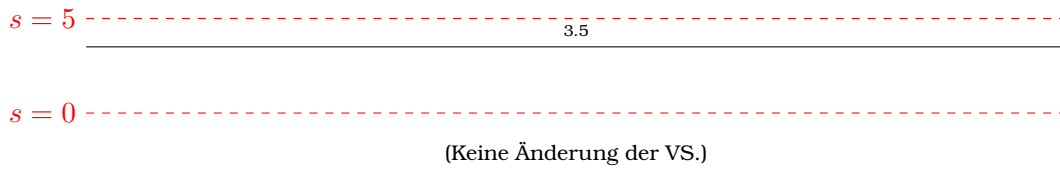


Abb. 4.8: 11. Tagesablauf: Tag mit konstanter Ventilstellung

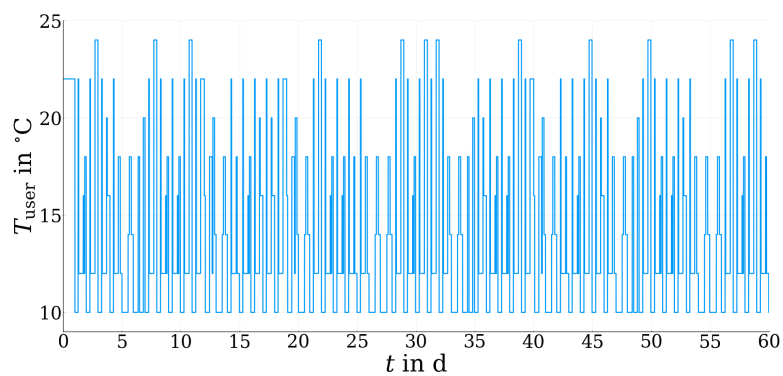


Abb. 4.9: Wunschttemperatur des Nutzers (Solltemperatur)

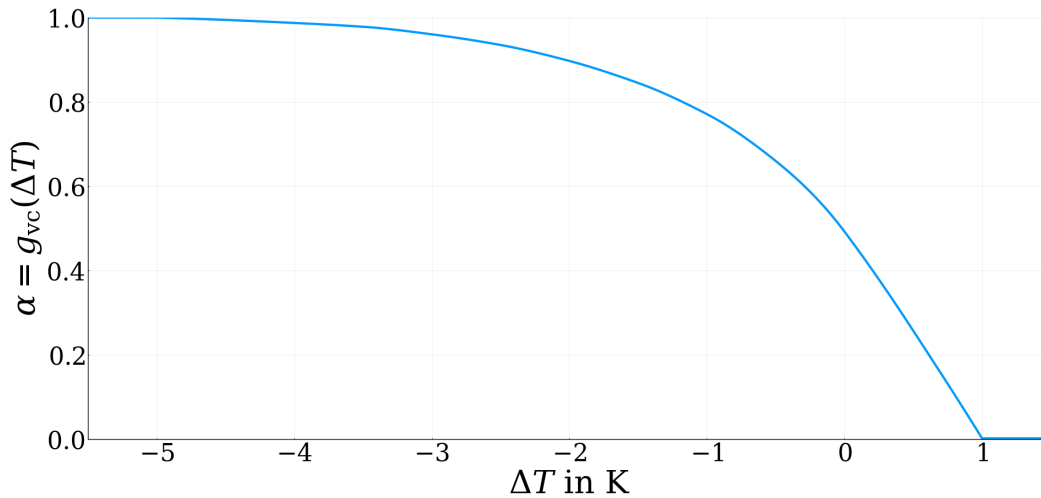


Abb. 4.10: Beispiel für eine Ventilkennlinie

gänzt durch einen ebenfalls von Heimeier produzierten, gasgefüllten Thermostatkopf, der bereits schließt, sobald $\Delta T = 1$ K ist. Für sechs spezielle Voreinstellungen dieses Ventils lassen sich aus einem Prospekt des Herstellers zu den entsprechenden Kennlinien gehörende Funktionswerte aus einem Diagramm ablesen [129, S. 7]. Die Tabelle B.2 im Anhang B.2 listet die entsprechenden Werte auf.

Es erweist sich als nützlich, die Ventilkennlinie mithilfe des Ventilöffnungsgrads $\alpha \in [0, 1]$ darzustellen. Die Größe α soll zu diesem Zweck das Verhältnis zwischen der jeweils aktuell durchströmten und der bei der ausgewählten Voreinstellung maximal durchströmten Querschnittsfläche angeben. Auf Basis der abgelesenen Werte aus dem Heimeier Prospekt ergibt sich daraus zum Beispiel eine Ventilkennlinie

$$\alpha := g_{vc}(T_{\text{room}} - T_{\text{user}}) \quad (39)$$

wie sie in Abbildung 4.10 zu sehen ist. Um zu verhindern, dass die Kurve durch Auftreten von kleinen Oszillationen unnatürlich erscheint, wurde für die Interpolation von einem Akima-Spline [3, S. 589 f.] Gebrauch gemacht, welcher sich aus stückweise kubischen Polynomen zusammensetzt und bei dem die erste Ableitung in einem Datenpunkt lokal zusammen mit den zwei linken und den zwei rechten Nachbarpunkten ermittelt wird. Außerdem wird für $\Delta T \geq 1$ K der theoretische Wert $\alpha = 0$ durch $\alpha = \alpha_{\min}$ mit $0 < \alpha_{\min} \ll 1$ [74, Folie 5] ersetzt, um einen kompletten Durchflussstop während der Simulation zu vermeiden. Eine Alternative zu der Akima-Spline Darstellung von g_{vc} , von der ebenfalls in

dieser Arbeit Gebrauch gemacht werden wird, ist die durch

$$g_{vc}(\Delta T) := \begin{cases} 1 - \exp(c_{vc,3} \cdot (\Delta T)^3 \\ \quad + c_{vc,2} \cdot (\Delta T)^2 + c_{vc,1} \cdot \Delta T + c_{vc,0}) & \text{für } \Delta T \leq 0 \text{ K,} \\ m_{vc} \cdot \Delta T + \alpha_{vc,0} & \text{für } 0 \text{ K} \leq \Delta T \leq 1 \text{ K,} \\ \alpha_{\min} & \text{für } 1 \text{ K} \leq \Delta T \end{cases} \quad (40)$$

definierte Näherung, die den Vorteil hat, dass sie wegen der Abhängigkeiten $\alpha_{vc,0} = 1 - \exp(c_{vc,0})$ und $m_{vc} \cdot 1 \text{ K} = \alpha_{\min} - \alpha_{vc,0} = \alpha_{\min} - (1 - \exp(c_{vc,0}))$, welche die Stetigkeit garantieren, nur vier Freiheitsgrade besitzt. Das Diagramm A.13 aus dem Anhang A.3 vergleicht die Approximation gemäß der Fallunterscheidung (40) mit dem Ergebnis des Akima-Splines. Der Ventilöffnungsgrad α erweist sich auch als nützlich, wenn es darum geht, die Druckdifferenz Δp_{valve} anzugeben, die dann entsteht, wenn das Ventil mit dem Volumenstrom \dot{V} durchströmt wird. Bei Kenntnis der charakteristischen Größe k_{vs} , welche dem Durchflusskoeffizienten des Ventils im Fall $\alpha = 1$ entspricht, kann der Druckabfall über dem Ventil durch

$$\Delta p_{\text{valve}} = \frac{\Delta p_{\text{ref}} \cdot \rho}{(k_{vs} \cdot \alpha)^2 \cdot \rho_{\text{ref}}} \cdot \dot{V}^2 \quad (41)$$

mit $\Delta p_{\text{ref}} = 10^5 \text{ Pa}$ und $\rho_{\text{ref}} = 1000 \frac{\text{kg}}{\text{m}^3}$ [4, S. 1838 f. (Abschn. 3.7.4)] berechnet werden.

Zur Modellierung des Betriebsverhaltens der Pumpe müssen sowohl das Kennfeld der Pumpe und damit ihre Drosselkurve als auch die hydraulischen Widerstände der Anlage und damit ihre Rohrleitungskennlinie bekannt sein. Der Parameter a_{sys} der Anlagenkennlinie (22) ist variabel, weil er vom hydraulischen Gesamtwiderstand des Systems abhängt und dieser sich durch das Öffnen oder Schließen des Thermostatventils verändert, wohingegen $H_{\text{sys},0}$ konstant ist. Weil der Ventilöffnungsgrad α die einfachste Möglichkeit bietet, um die Dynamik der Anlagenkennlinie mathematisch zu beschreiben, wird im Julia-Modell auf Grundlage der hydraulischen Widerstände, speziell die der im Abschnitt 3.1 vorgestellten Rohre, die Funktion $a_{\text{sys}}(\alpha)$ definiert. Für die Darstellung aller Pumpenkennlinien reichen nach Abschnitt 3.3 bereits N Messwertpaare $((\dot{V}_{\text{ref}})_i, (H_{\text{ref}})_i)$ einer Referenz-Kennlinie aus, die hier nach aufsteigendem Volumenstrom \dot{V} sortiert seien. Für diese gelte außerdem das Monotonieverhalten $(H_{\text{ref}})_i < (H_{\text{ref}})_{i+1}$ für $i = 1, \dots, N - 1$, $(H_{\text{ref}})_N = 0.00 \text{ m}$ sowie $(\dot{V}_{\text{ref}})_1 = 0.0 \frac{\text{m}^3}{\text{s}}$. Die Drehzahl n_{ref} wird dabei nach Möglichkeit so gewählt, dass die zugehörige Pumpenkennlinie sich vorzugsweise mittig zwischen den Kennlinien der kleinsten von der Pumpe ansteuerbaren Drehzahl n_{min} und der größten n_{max} befindet. Damit kann die Drosselkurve der Referenzdrehzahl durch stückweise lineare Interpolation [58, Abschn. 3.1] der Messwertpaare auf dem

Intervall $[(\dot{V}_{\text{ref}})_i, (\dot{V}_{\text{ref}})_{i+1}]$ als

$$H_{n_{\text{ref}}}(\dot{V}) := (H_{\text{ref}})_i + \frac{(H_{\text{ref}})_{i+1} - (H_{\text{ref}})_i}{(\dot{V}_{\text{ref}})_{i+1} - (\dot{V}_{\text{ref}})_i} \cdot (\dot{V} - (\dot{V}_{\text{ref}})_i) \quad (42)$$

dargestellt werden. Um die Drosselkurve einer unbekanntenen, anderen Drehzahl $n_{\text{u}} \in [n_{\text{min}}, n_{\text{max}}]$ zu ermitteln, wird auf die Affinitätsgesetze (19) zurückgegriffen, aus denen

$$H_{n_{\text{u}}}(\dot{V}) := r^2 \cdot H_{n_{\text{ref}}}\left(\frac{1}{r} \cdot \dot{V}\right) \quad \text{mit } r := \frac{n_{\text{u}}}{n_{\text{ref}}} \quad (43)$$

folgt. Falls $n_{\text{pump}}(t)$ die derzeitige Drehzahl der Pumpe und $a_{\text{sys}}(t)$ sowie $H_{\text{sys},0}(t)$ die derzeitigen Parameter für die Approximation der Anlagenkennlinie nach Formel (22) sind, dann berechnet sich der Volumenstrom des aktuellen Betriebspunktes der Pumpe unter den vorangegangenen Bedingungen auf folgende Weise:

$$g_{\text{opp}}(n_{\text{pump}}, a_{\text{sys}}(\alpha)) := \frac{m_{\text{pc}}}{2 \cdot a_{\text{sys}}} + \sqrt{\left(\frac{m_{\text{pc}}}{2 \cdot a_{\text{sys}}}\right)^2 + \frac{H_{\text{pc}} - H_{\text{sys},0}}{a_{\text{sys}}}}. \quad (44)$$

Die hier verwendeten Abkürzungen leiten sich aus den Formeln (42) und (43) her und sind gegeben durch

$$m_{\text{pc}} := \frac{n_{\text{pump}}}{n_{\text{ref}}} \cdot \frac{(H_{\text{ref}})_{j+1} - (H_{\text{ref}})_j}{(\dot{V}_{\text{ref}})_{j+1} - (\dot{V}_{\text{ref}})_j} \quad \text{und}$$

$$H_{\text{pc}} := -m_{\text{pc}} \cdot \frac{n_{\text{pump}}}{n_{\text{ref}}} \cdot (\dot{V}_{\text{ref}})_j + \left(\frac{n_{\text{pump}}}{n_{\text{ref}}}\right)^2 \cdot (H_{\text{ref}})_j,$$

wobei der wegen der Monotonie der Pumpenkennlinie eindeutig bestimmte Index j die Ungleichungen

$$\left(\frac{n_{\text{pump}}}{n_{\text{ref}}}\right)^2 \cdot (H_{\text{ref}})_j > a_{\text{sys}} \cdot \left(\frac{n_{\text{pump}}}{n_{\text{ref}}} \cdot (\dot{V}_{\text{ref}})_j\right)^2 + H_{\text{sys},0}$$

$$\left(\frac{n_{\text{pump}}}{n_{\text{ref}}}\right)^2 \cdot (H_{\text{ref}})_{j+1} \leq a_{\text{sys}} \cdot \left(\frac{n_{\text{pump}}}{n_{\text{ref}}} \cdot (\dot{V}_{\text{ref}})_{j+1}\right)^2 + H_{\text{sys},0}$$

erfüllt. Demgegenüber zeigt der Quelltext aus Anhang (D.1) wie der Volumenstrom des aktuellen Betriebspunktes berechnet werden kann, falls die Anlagenkennlinie nach Gleichung (21) modelliert wird.

Wenn die Relationen (31), (32), (33), (35), (36) und (37) der entsprechenden Wärmeströme in den differentiellen Block (38) eingesetzt werden, entsteht zusammen mit den drei algebraischen Gleichungen (24), (39) und (44) insgesamt das folgende DAE-System [40, Folie 9–13]. Dieses wurde im Rahmen des

SOPHi-Projektes ausschließlich vom Autor der vorliegenden Arbeit formuliert und geht auf einen Ansatz von Frischmuth [36, Folien 16–20] zurück.

$$\begin{aligned}
 \dot{T}_{\text{room}}(t) &= \frac{1}{C_{\text{room}}} \cdot \left(\nu_{\text{wall,int}}(T_{\text{wall}}(t) - T_{\text{room}}(t)) + \nu_{\text{fc,int}}(T_{\text{fc}}(t) - T_{\text{room}}(t)) \right. \\
 &\quad + \nu_{\text{win}}(T_{\text{out}}(t) - T_{\text{room}}(t)) + \nu_{\text{htr}}(T_{\text{htr}}(t) - T_{\text{room}}(t))^{n_{\text{htr}}} \\
 &\quad \left. + \nu_{\text{htr}}(T_{\text{rtn}}(t) - T_{\text{room}}(t))^{n_{\text{htr}}} + \dot{Q}_{\text{add}}(t) \right) \\
 \dot{T}_{\text{wall}}(t) &= \frac{1}{C_{\text{wall}}} \cdot \left(\nu_{\text{wall,int}}(T_{\text{room}}(t) - T_{\text{wall}}(t)) + \nu_{\text{wall,ext}}(T_{\text{out}}(t) - T_{\text{wall}}(t)) \right) \\
 \dot{T}_{\text{fc}}(t) &= \frac{1}{C_{\text{fc}}} \cdot \left(\nu_{\text{fc,int}}(T_{\text{room}}(t) - T_{\text{fc}}(t)) + \nu_{\text{fc,ext}}(T_{\text{out}}(t) - T_{\text{fc}}(t)) \right) \\
 \dot{T}_{\text{htr}}(t) &= \frac{1}{C_{\text{htr}}} \cdot \left(-\nu_{\text{htr}}(T_{\text{htr}}(t) - T_{\text{room}}(t))^{n_{\text{htr}}} + \nu_{\text{w}} \cdot \dot{V}(t) \cdot (T_{\text{flw}}(t) - T_{\text{htr}}(t)) \right) \\
 \dot{T}_{\text{rtn}}(t) &= \frac{1}{C_{\text{htr}}} \cdot \left(-\nu_{\text{htr}}(T_{\text{rtn}}(t) - T_{\text{room}}(t))^{n_{\text{htr}}} + \nu_{\text{w}} \cdot \dot{V}(t) \cdot (T_{\text{htr}}(t) - T_{\text{rtn}}(t)) \right) \\
 0 &= T_{\text{flw}}(t) - g_{\text{hc}}(T_{\text{out}}(t)) \\
 0 &= \alpha(t) - g_{\text{vc}}(T_{\text{room}}(t) - T_{\text{user}}(t)) \\
 0 &= \dot{V}(t) - g_{\text{opp}}(n_{\text{pump}}(t), a_{\text{sys}}(\alpha(t))) \tag{45}
 \end{aligned}$$

Da die drei algebraischen Gleichungen explizit nach T_{flw} , α und \dot{V} auflösbar sind, ist das System durch Ersetzen dieser drei Größen mit den entsprechenden Ausdrücken in ein reines, von den algebraischen Gleichungen freies ODE-System überführbar. Mit Ausnahme des nächsten Kapitels 5, in welchem es thematisch um die Systemidentifikation geht, wird von dieser Möglichkeit im restlichen Teil der Arbeit Gebrauch gemacht werden. Die Quelltextauszüge in den Anhängen D.3 und D.4 beinhalten die Implementierungen der beiden mathematisch äquivalenten Varianten des Julia-Modells als DAE-System und als ODE-System.

4.3 Modellvergleich

Um das Benchmark-Modell, das in Dymola erstellt wurde, mit dem zu vergleichen, das mit Julia erstellt wurde, müssen beide auf dieselbe Weise parametrisiert werden. Für alle nachfolgenden Untersuchungen reicht es aus, den festen Satz von Parametern aus Anhang C.3 vorzugeben und mit den Anfangsbedingungen aus Anhang C.4 auf dem Intervall $[t_0, t_f]$ zu arbeiten. Speziell für den Modellvergleich wird dabei die Drehzahl der Pumpe konstant gehalten, also $n_{\text{pump}}(t) \equiv n_{\text{pump},0}$ benutzt. Die im DAE-System für das Julia-Modell auftauchenden Parameter sind die Kombination von teilweise mehreren Parametern des Dymola-Modells, auf dessen genaue Angabe in Form von konkreten nume-

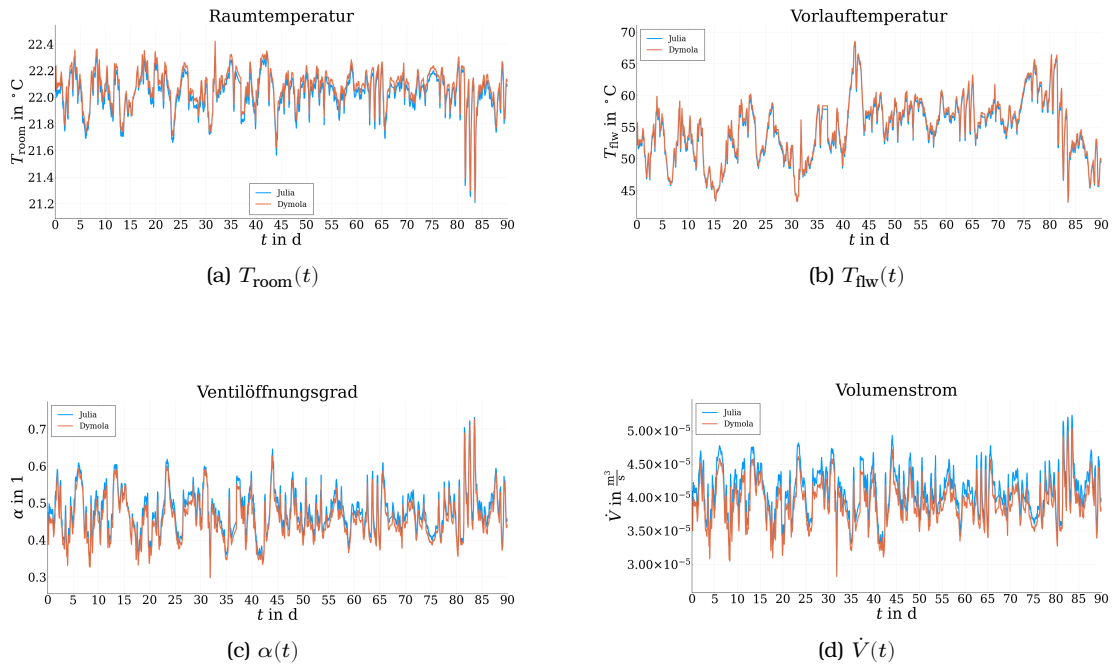


Abb. 4.11: Vergleich zwischen den Simulationsergebnissen aus Dymola und den durch Julia ermittelten Lösungen des DAE-Systems (45) für das Benchmark-Modell [40, Folie 15–18]

rischen Werten aufgrund der großen Anzahl hier verzichtet wird. Dem interessierten Leser steht jedoch auf dem Speichermedium im Anhang das vollständig parametrisierte Dymola-Modell zur Verfügung. Zudem befinden sich im Anhang D.2 die entsprechenden Julia-Quelltexte, welche sich auf die Umrechnung der Parameter beziehen.

Neben der Anzahl der Parameter ist auch die der Gleichungen im Dymola-Modell mit insgesamt 1591 wesentlich größer. Einer der Gründe dafür ist, dass in dem Julia-Modell beispielsweise für die Wand nur die Zustandsgröße T_{wall} vorkommt, wohingegen in Modelica für alle vier Raumwände einzeln jeweils die drei Temperaturen aus den Formeln (27)–(30) herangezogen werden. Ein weiterer Grund sind mitunter triviale Gleichungen für die Flussgrößen der Konnektoren, die auf Basis des Knotensatzes (9) von Modelica erzeugt werden. Obwohl das DAE-System (45), das mithilfe von Julia gelöst wird, aus lediglich acht Gleichungen besteht, belegt die Abbildung 4.11, dass beide Modelle nahezu identische Resultate liefern. Speziell bei den für den weiteren Verlauf der kommenden Untersuchungen entscheidenden Zustandsgrößen, der Raumtemperatur, des Ventilöffnungsgrads und des Volumenstroms, gibt es in den Simulationsergebnissen keine Unterschiede bei den ersten beiden Nachkommastellen. Die CPU-Zeit für die Berechnung aller Ergebnisse liegt für eine

HP Z4 G4 Workstation (Intel(R) Core(TM) i9-109000X Prozessor, CPU @ 3.70 GHz, 10 Kerne, 20 logische Prozessoren, 64 GB RAM, Windows 11 Pro Version 24H2) bei einer Mittelung von zehn Simulationsdurchläufen mit dem Dymola-Modell (Dymola Version 2025x vom 11.10.2024 + Visual Studio Community 2022 Version 17.13.6) bei ca. 8.29 Sekunden und ist damit etwa 47-mal länger als die CPU-Zeit auf derselben Workstation mit dem Julia-Modell (Julia Version 1.11.5 vom 14.04.2025), das im Mittel nur 0.175 Sekunden für die gleiche Aufgabe braucht. Der gewaltige Unterschied erklärt sich zum einen dadurch, dass Julia eine sehr effiziente Programmiersprache ist (siehe Unterabschnitt 4.1.3), und zum anderen an der deutlich höheren Anzahl an Gleichungen, die das Dymola-Modell intern aufstellt (siehe Unterabschnitt 4.1.1).

Aber nicht nur durch die viel schnelleren Rechenzeiten muss die Wahl in Bezug auf den Nutzungszweck in dieser Arbeit auf das Julia-Modell fallen. Auch das akausale Programmiergrundkonzept von Modelica, das weniger Möglichkeiten für das Aufstellen und Lösen von Optimierungsproblemen und das Einbinden von Steuerstrategien bietet, sowie die Gegenüberstellung bereits vorhandener Bibliotheken sprechen für die Wahl von Julia.

5 Systemidentifikation

Dieses Kapitel startet mit einer theoretischen Analyse, bei der mathematische Methoden zur Identifikation der Parameter des DAE-Systems (45) vorgestellt werden. Die Beobachtungen, die sich aus den numerischen Untersuchungen ergeben, werden anschließend im Kontext starkerer, fur die praktische Anwendung relevanten Voraussetzungen interpretiert.

5.1 Identifikation

Die mathematische Analyse dieses Abschnitts wurde vom Autor dieser Arbeit zum Teil bereits im Rahmen des SOPHi-Projektes durchgefuhrt [32, S. 41–48] und basiert auf den numerischen Methoden, die in einem Review-Paper von Bock, Kostina und Schloder [20, S. 367–381] zusammengetragen wurden.

Ausgehend von der vektoriellen Darstellung

$$\begin{aligned} \mathbf{y}(t) &:= (T_{\text{room}}(t) \quad T_{\text{wall}}(t) \quad T_{\text{fc}}(t) \quad T_{\text{htr}}(t) \quad T_{\text{rtn}}(t))^{\text{T}} \\ \mathbf{y}_0 &:= (T_{\text{room},0} \quad T_{\text{wall},0} \quad T_{\text{fc},0} \quad T_{\text{htr},0} \quad T_{\text{rtn},0})^{\text{T}} \\ \mathbf{z}(t) &:= (T_{\text{flw}}(t) \quad \alpha(t) \quad \dot{V}(t))^{\text{T}} \\ \mathbf{z}_0 &:= (T_{\text{flw},0} \quad \alpha_0 \quad \dot{V}_0)^{\text{T}} \\ \mathbf{x}(t) &:= (\mathbf{y}(t)^{\text{T}} \quad \mathbf{z}(t)^{\text{T}})^{\text{T}} \end{aligned} ,$$

bei der die differentiellen und algebraischen Zustandsgroen aus (34) separiert werden, lasst sich das DAE-System aus dem Unterabschnitt 4.2.2 umformulieren. Nach Vereinigung der Storgroen zum Vektor $\mathbf{d}(t)$, Einfuhrung der Steuergroe $u(t)$ und Zusammenfassung aller Parameter der differentiellen Gleichungen $\boldsymbol{\theta}_{\text{d}}$ sowie aller Parameter der algebraischen Gleichungen $\boldsymbol{\theta}_{\text{a}}$ zu $\boldsymbol{\theta}$ gema

$$\begin{aligned} \mathbf{d}(t) &:= (T_{\text{out}}(t) \quad T_{\text{user}}(t) \quad Q_{\text{add}}(t))^{\text{T}} \\ u(t) &:= n_{\text{pump}}(t) \\ \boldsymbol{\theta}_{\text{d}} &:= (n_{\text{htr}} \quad C_{\text{room}} \quad C_{\text{wall}} \quad C_{\text{fc}} \quad C_{\text{htr}} \quad \nu_{\text{wall,int}} \\ &\quad \nu_{\text{wall,ext}} \quad \nu_{\text{fc,int}} \quad \nu_{\text{fc,ext}} \quad \nu_{\text{win}} \quad \nu_{\text{htr}} \quad \nu_{\text{w}})^{\text{T}} \\ \boldsymbol{\theta}_{\text{a}} &:= (\boldsymbol{\theta}_{\text{hc}}^{\text{T}} \quad \boldsymbol{\theta}_{\text{vc}}^{\text{T}} \quad \boldsymbol{\theta}_{\text{opp}}^{\text{T}})^{\text{T}} \quad \boldsymbol{\theta} := (\boldsymbol{\theta}_{\text{d}}^{\text{T}} \quad \boldsymbol{\theta}_{\text{a}}^{\text{T}})^{\text{T}} \end{aligned}$$

nehmen die dafür benötigten Funktionen f und g die folgende Gestalt an:

$$\mathbf{f}(t, \mathbf{y}, \mathbf{z}, \mathbf{d}, u; \boldsymbol{\theta}_d) := \begin{pmatrix} \frac{1}{\theta_{d,2}} \cdot (\theta_{d,6}(y_2 - y_1) + \theta_{d,8}(y_3 - y_1) + \theta_{d,10}(d_1 - y_1) + \theta_{d,11}(y_4 - y_1)^{\theta_{d,1}} + \theta_{d,11}(y_5 - y_1)^{\theta_{d,1}} + d_3) \\ \frac{1}{\theta_{d,3}} \cdot (\theta_{d,6}(y_1 - y_2) + \theta_{d,7}(d_1 - y_2)) \\ \frac{1}{\theta_{d,4}} \cdot (\theta_{d,8}(y_1 - y_3) + \theta_{d,9}(d_1 - y_3)) \\ \frac{1}{\theta_{d,5}} \cdot (-\theta_{d,11}(y_4 - y_1)^{\theta_{d,1}} + \theta_{d,12} \cdot z_3 \cdot (z_1 - y_4)) \\ \frac{1}{\theta_{d,5}} \cdot (-\theta_{d,11}(y_5 - y_1)^{\theta_{d,1}} + \theta_{d,12} \cdot z_3 \cdot (y_4 - y_5)) \end{pmatrix},$$

$$\mathbf{g}(t, \mathbf{y}, \mathbf{z}, \mathbf{d}, u; \boldsymbol{\theta}_a) := \begin{pmatrix} z_1 - g_{\text{hc}}(d_1; \boldsymbol{\theta}_{\text{hc}}) \\ z_2 - g_{\text{vc}}(y_1 - d_2; \boldsymbol{\theta}_{\text{vc}}) \\ z_3 - g_{\text{opp}}(u, a_{\text{sys}}(z_2); \boldsymbol{\theta}_{\text{opp}}) \end{pmatrix}.$$

Weil in diesem Kapitel für g_{hc} und g_{vc} aufgrund der kleineren Anzahl an Parametern die Näherungen aus (25) und (40) benutzt werden sollen, ergibt sich

$$\boldsymbol{\theta}_{\text{hc}} := (m_{\text{hc}} \quad T_{\text{hc},0})^T \quad \text{und} \quad \boldsymbol{\theta}_{\text{vc}} := (c_{\text{vc},3} \quad c_{\text{vc},2} \quad c_{\text{vc},1} \quad c_{\text{vc},0} \quad \alpha_{\text{min}})^T.$$

Das Benchmark-Modell in Form des DAE-Systems (45) mit den Anfangsbedingungen aus Anhang C.4 ist bei Verwendung der obigen Notation folglich äquivalent zu

$$\begin{aligned} \dot{\mathbf{y}}(t) &= \mathbf{f}(t, \mathbf{y}(t), \mathbf{z}(t), \mathbf{d}(t), u(t); \boldsymbol{\theta}_d) \quad \text{für } t \in [t_0, t_f^{\text{id}}] \\ 0 &= \mathbf{g}(t, \mathbf{y}(t), \mathbf{z}(t), \mathbf{d}(t), u(t); \boldsymbol{\theta}_a) \quad \text{für } t \in [t_0, t_f^{\text{id}}] \\ \mathbf{x}(t_0) &= \mathbf{x}_0. \end{aligned} \tag{46}$$

Für die im Folgenden vorgestellten Methoden zur Identifikation der Systemparameter $\boldsymbol{\theta}$, reicht es aus, sich auf

$$\begin{aligned} \boldsymbol{\theta}^{\text{id}} &:= (\theta_{d,1} \quad \theta_{d,2} \quad \theta_{d,3} \quad \theta_{d,4} \quad \theta_{d,5} \quad \theta_{d,6} \quad \theta_{d,7} \quad \theta_{d,8} \\ &\quad \theta_{d,9} \quad \theta_{d,10} \quad \theta_{d,11} \quad \theta_{\text{hc},1} \quad \theta_{\text{hc},2} \quad \theta_{\text{vc},1} \quad \theta_{\text{vc},2} \quad \theta_{\text{vc},3} \quad \theta_{\text{vc},4})^T \\ &= (n_{\text{htr}} \quad C_{\text{room}} \quad C_{\text{wall}} \quad C_{\text{fc}} \quad C_{\text{htr}} \quad \nu_{\text{wall,int}} \quad \nu_{\text{wall,ext}} \quad \nu_{\text{fc,int}} \\ &\quad \nu_{\text{fc,ext}} \quad \nu_{\text{win}} \quad \nu_{\text{htr}} \quad m_{\text{hc}} \quad T_{\text{hc},0} \quad c_{\text{vc},3} \quad c_{\text{vc},2} \quad c_{\text{vc},1} \quad c_{\text{vc},0})^T \end{aligned} \tag{47}$$

zu beschränken und nur diese Parameter als unbekannt vorauszusetzen. Die Gründe dafür lauten zum einen, dass der Wert von $\nu_{\text{w}} = c_{\text{w}} \cdot \rho_{\text{w}}$ als Produkt von zwei Konstanten selbst konstant ist sowie, dass der Wert von α_{min} ebenfalls

fest vorgegeben ist. Zum anderen ist der aktuelle Betriebspunkt der Pumpe zu jedem Zeitpunkt aufgrund der in der Pumpe hinterlegten Kennfelder in der Praxis bekannt, was mit der Kenntnis der Parameter θ_{opp} , die ausschließlich zur Berechnung des Funktionswertes von g_{opp} benötigt werden, korrespondiert. Zur Bestimmung von θ^{id} wird davon ausgegangen, dass für eine Auswahl $I_{\text{msr}} \subseteq I_{\text{all}} := \{1, 2, \dots, 8\}$ mit $N_{\text{msr}} := |I_{\text{msr}}|$ der acht Zustandsgrößen aus (34) „Messwerte“ zur Verfügung stehen. Die Zeitpunkte $t_j^{\text{id}} \in [t_0, t_f^{\text{id}}]$ für $j = 1, \dots, M^{\text{id}}$, zu denen die Werte aller messbaren Größen x_i mit $i \in I_{\text{msr}}$ gleichzeitig aufgenommen werden, seien im äquidistanten Abstand von Δt_μ^{id} definiert durch $t_j^{\text{id}} := t_0 + j \cdot \Delta t_\mu^{\text{id}}$. Da die theoretischen Untersuchungen in dieser Arbeit nicht auf tatsächlich durchgeführten Messungen an einem realen Heizsystem beruhen, ergeben sich die „Messwerte“ aus der Simulation des DAE-Modells (46) mit den vorgegebenen Modellparametern aus Anhang C.3. Sei also die Matrix μ aller Messwerte mit der Dimension $N_{\text{msr}} \times M^{\text{id}}$ definiert durch die theoretischen Werte der Lösung von (46), das heißt $\mu_{ij} := x_i(t_j^{\text{id}})$.

Das prinzipielle Vorgehen zur Ermittlung der unbekannt angenommenen Parameter θ^{id} sieht so aus, dass mit $K^{\text{id}} + 1$ geeigneten, aufsteigend sortierten Zeitpunkten τ_k^{id} im äquidistanten Abstand von $\Delta \tau^{\text{id}}$ und mit $\tau_0^{\text{id}} = t_0$ sowie $\tau_{K^{\text{id}}}^{\text{id}} = t_f^{\text{id}}$ auf jedem der K^{id} Intervalle $[\tau_k^{\text{id}}, \tau_{k+1}^{\text{id}}]$ ein durch $\delta^{(k)}$ relaxiertes DAE-System aufgestellt wird, dessen Anfangswerte $\beta^{(k)}$ in Form eines Mehrfachschießverfahrens bei der Formulierung eines Optimierungsproblems genutzt werden. Mit den monoton fallenden Funktionen

$$\delta^{(k)}(t) := \begin{cases} 1 - \frac{t - \tau_k^{\text{id}}}{\tau_{k+1}^{\text{id}} - \tau_k^{\text{id}}} & t \in [\tau_k^{\text{id}}, \tau_{k+1}^{\text{id}}], \\ 0 & t \notin [\tau_k^{\text{id}}, \tau_{k+1}^{\text{id}}], \end{cases}$$

welche die Eigenschaften $\delta^{(k)}(t) \geq 0$ für alle $t \in \mathbb{R}$ sowie $\delta^{(k)}(\tau_k^{\text{id}}) = 1$ erfüllen und zur Verbesserung der Konsistenz der algebraischen Bedingungen beitragen, lauten die relaxierten DAE-Systeme

$$\begin{aligned} \dot{\mathbf{y}}_{\text{r}}^{(k)}(t) &= \mathbf{f}(t, \mathbf{y}_{\text{r}}^{(k)}(t), \mathbf{z}_{\text{r}}^{(k)}(t), \mathbf{d}(t), u(t); \boldsymbol{\theta}_{\text{d}}) && \text{für } t \in [\tau_k^{\text{id}}, \tau_{k+1}^{\text{id}}] \\ \mathbf{0} &= \mathbf{g}(t, \mathbf{y}_{\text{r}}^{(k)}(t), \mathbf{z}_{\text{r}}^{(k)}(t), \mathbf{d}(t), u(t); \boldsymbol{\theta}_{\text{a}}) \\ &\quad - \delta^{(k)}(t) \cdot \mathbf{g}(\tau_k^{\text{id}}, \boldsymbol{\beta}_{1..5}^{(k)}, \boldsymbol{\beta}_{6..8}^{(k)}, \mathbf{d}(\tau_k^{\text{id}}), u(\tau_k^{\text{id}}); \boldsymbol{\theta}_{\text{a}}) && \text{für } t \in [\tau_k^{\text{id}}, \tau_{k+1}^{\text{id}}] \\ \mathbf{y}_{\text{r}}^{(k)}(\tau_k^{\text{id}}) &= \boldsymbol{\beta}_{1..5}^{(k)}, \quad \mathbf{z}_{\text{r}}^{(k)}(\tau_k^{\text{id}}) = \boldsymbol{\beta}_{6..8}^{(k)}. \end{aligned} \tag{48}$$

Nach Bestimmung der Lösungen $\mathbf{x}_{\text{r}}^{(k)}(t) := (\mathbf{y}_{\text{r}}^{(k)}(t)^{\text{T}} \quad \mathbf{z}_{\text{r}}^{(k)}(t)^{\text{T}})^{\text{T}}$ erfolgt die Konstruktion der Funktion $\mathbf{x}_{\text{r}}(t)$ derart, dass für alle $k = 0, \dots, K^{\text{id}} - 1$ und für alle $t \in [\tau_k^{\text{id}}, \tau_{k+1}^{\text{id}}]$ die Beziehung $\mathbf{x}_{\text{r}}(t) := \mathbf{x}_{\text{r}}^{(k)}(t)$ gilt. Weil den Anfangswerten $\boldsymbol{\beta}^{(0)} := \mathbf{x}_0$ die Anfangsbedingungen aus Anhang C.4 zugeordnet werden, bezieht sich das nachfolgende Mehrfachschießverfahren nur auf die Anfangswerte

te $\beta^{(k)}$ mit $k > 0$. Die gesuchten Parameter θ^{id} bilden zusammen mit den letztgenannten Größen $(\beta^{(k)})_{k=1, \dots, K^{\text{id}}-1}$ die Lösungsvariablen eines Optimierungsproblems. Dessen Zielfunktion J^{id} gibt die Summe der quadratischen Abweichungen der aus den Lösungen der relaxierten DAE-Systeme (48) zusammengesetzten Funktion $x_r(t)$ zu den Messwerten μ_{ij} an, wobei die einzelnen Summanden durch die Gewichtungsfaktoren $1/\sigma_{ij}^2$ für $i \in I_{\text{msr}}$ und $j = 1, \dots, M^{\text{id}}$ den Größenordnungen der Messwerte entsprechend skaliert werden. Die Minimierung der Zielfunktion J^{id} erfolgt unter den Anpassungsbedingungen $G_1^{\text{id},(k)}$ für $k = 0, \dots, K^{\text{id}} - 2$, welche die Stetigkeit von $x_r(t)$ garantieren sowie den algebraischen Bedingungen $G_2^{\text{id},(k)}$ für $k = 0, \dots, K^{\text{id}} - 1$, die sicherstellen, dass auch die ursprünglichen algebraischen Gleichungen erfüllt bleiben.

$$\begin{aligned}
 J^{\text{id}}(\theta^{\text{id}}, (\beta^{(k)})_{k=1, \dots, K^{\text{id}}-1}) &:= \sum_{i \in I_{\text{msr}}} \sum_{j=1}^{M^{\text{id}}} \frac{(\mu_{ij} - x_{r,i}(t_j^{\text{id}}, \theta^{\text{id}}, (\beta^{(k)})_{k=1, \dots, K^{\text{id}}-1}))^2}{\sigma_{ij}^2} \rightarrow \min \\
 G_1^{\text{id},(k)}(\theta^{\text{id}}, \beta^{(k)}) &:= \mathbf{y}_r^{(k)}(\tau_{k+1}^{\text{id}}; \beta^{(k)}, \theta_d) - \beta_{1..5}^{(k+1)} = \mathbf{0} \quad \text{für } k = 0, \dots, K^{\text{id}} - 2 \\
 G_2^{\text{id},(k)}(\theta^{\text{id}}, \beta^{(k)}) &:= \mathbf{g}(\tau_k^{\text{id}}, \beta_{1..5}^{(k)}, \beta_{6..8}^{(k)}, \mathbf{d}(\tau_k^{\text{id}}), u(\tau_k^{\text{id}}); \theta_a) = \mathbf{0} \quad \text{für } k = 0, \dots, K^{\text{id}} - 1
 \end{aligned} \tag{49}$$

5.2 Ergebnisse der Identifikation

Im ersten Schritt bietet es sich an, herauszufinden, ob eine prinzipielle Identifikation der unbekannt Systemparameter aus (47) überhaupt möglich ist. Dazu wird von dem Idealfall ausgegangen, dass alle acht in Formel (34) gelisteten Zustandsgrößen mit hoher Genauigkeit messbar sind. Die entsprechende Bedingung lautet $I_{\text{msr}} = I_{\text{all}}$. Auch wenn eine solche Voraussetzung in der Praxis bestenfalls in speziell ausgestatteten Testgebäuden erfüllbar wäre, ist die Untersuchung dieses Falls im Hinblick auf theoretische Aussagen zur Identifizierbarkeit des Systems interessant.

Die Durchführung der Optimierung (49) zur Bestimmung einer Lösung $\theta_{\text{res}}^{\text{id}}$ erfolgt unter der Annahme, dass

$$\theta_1^{\text{id}} \in [1.25, 1.33]$$

gilt [4, S. 791 (Tafel 2.2.2-2)]. Für alle übrigen als unbekannt vorausgesetzten Parameter wird zudem gefordert, dass sie jeweils auf einen Wert, der betragsmäßig maximal 25 % von dem gesuchten Wert abweicht, geschätzt werden

5 Systemidentifikation

$\theta_{\text{res},1}^{\text{id}}$	$\theta_{\text{res},2}^{\text{id}}$	$\theta_{\text{res},3}^{\text{id}}$	$\theta_{\text{res},4}^{\text{id}}$	$\theta_{\text{res},5}^{\text{id}}$	$\theta_{\text{res},6}^{\text{id}}$	$\theta_{\text{res},7}^{\text{id}}$	$\theta_{\text{res},8}^{\text{id}}$	$\theta_{\text{res},9}^{\text{id}}$
-0.8 %	-1.8 %	-1.9 %	-2.9 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %
$\theta_{\text{res},10}^{\text{id}}$	$\theta_{\text{res},11}^{\text{id}}$	$\theta_{\text{res},12}^{\text{id}}$	$\theta_{\text{res},13}^{\text{id}}$	$\theta_{\text{res},14}^{\text{id}}$	$\theta_{\text{res},15}^{\text{id}}$	$\theta_{\text{res},16}^{\text{id}}$	$\theta_{\text{res},17}^{\text{id}}$	
0.0 %	0.0 %	0.0 %	-0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	

Tab. 5.1: Prozentuale Abweichungen der besten Näherung $\theta_{\text{res}}^{\text{id}}$ von den tatsächlichen Systemparametern im Fall $I_{\text{msr}} = I_{\text{all}}$

können, sodass die Einschränkungen

$$\begin{aligned}
 |\theta_i^{\text{id}}| &\in \left[0.75 \cdot |\theta_{\text{d},i}|, 1.25 \cdot |\theta_{\text{d},i}| \right] \text{ für } i \in \{2, \dots, 11\}, \\
 |\theta_{11+i}^{\text{id}}| &\in \left[0.75 \cdot |\theta_{\text{hc},i}|, 1.25 \cdot |\theta_{\text{hc},i}| \right] \text{ für } i \in \{1, 2\} \text{ und} \\
 |\theta_{13+i}^{\text{id}}| &\in \left[0.75 \cdot |\theta_{\text{vc},i}|, 1.25 \cdot |\theta_{\text{vc},i}| \right] \text{ für } i \in \{1, \dots, 4\}
 \end{aligned}$$

zutreffen. Außerdem werden für die Optimierung zufällige Startwerte für alle θ_i^{id} aus den entsprechenden Intervallen gewählt und für $i \in \{2, \dots, 17\}$ sogar nur solche, die betragsmäßig allesamt mindestens 15% von ihren gesuchten Werten abweichen, um so ein Szenario nachzustellen, das von einer recht ungünstigen anfänglichen Initialisierung ausgeht. Des Weiteren sei für die Steuergröße $u(t)$ während des gesamten Zeitraums $[t_0, t_f^{\text{id}}]$ durch $u(t) \equiv n_{\text{pump},0}$ eine konstante Drehzahl vorgegeben. Mit Festlegung der Parameterwerte aus Anhang C.5, die einen Modellzeitraum von einer Woche mit stündlich verfügbaren Messwerten beschreiben, wird die Charakterisierung der Problemstellung schließlich komplementiert.

Zur Lösung des vorliegenden Optimierungsproblems wurden eine Reihe von Solvern aus den entsprechenden Standardbibliotheken von Julia ausprobiert, doch nur mit einem Wrapper des Algorithmus Ipopt [131] konnten brauchbare Ergebnisse in einer angemessenen Rechenzeit von wenigen Minuten erzielt werden. Die Tabelle 5.1 zeigt die prozentualen Abweichungen der berechneten Lösung $\theta_{\text{res}}^{\text{id}}$ von Ipopt zu den tatsächlichen Werten der Systemparameter, welche im Anhang C.3 zu finden sind.

Marktübliche Heizungsumwälzpumpen verfügen aus wirtschaftlichen Gründen über deutlich weniger Informationen als im bisher betrachteten theoretischen Fall $I_{\text{msr}} = I_{\text{all}}$ angenommen [50, Folie 5]. So weiß eine Heizungsumwälzpumpe bloß welche Drehzahl $n(t)$ sie fährt, wie viel elektrische Leistung sie benötigt, um den Motor anzutreiben und wie groß ihre Wicklungstemperatur ist. Durch die Hinterlegung eines Leistungskennfeldes kann mithilfe der elektrischen Leistung ein Wert für $\dot{V}(t)$ geschätzt werden und damit wiederum über ein ebenfalls hinterlegtes hydraulisches Kennfeld auch ein Wert für $H(t)$. Außerdem bietet die Wicklungstemperatur die Chance an eine Approximation

5 Systemidentifikation

$\theta_{\text{res},1}^{\text{id}}$	$\theta_{\text{res},2}^{\text{id}}$	$\theta_{\text{res},3}^{\text{id}}$	$\theta_{\text{res},4}^{\text{id}}$	$\theta_{\text{res},5}^{\text{id}}$	$\theta_{\text{res},6}^{\text{id}}$	$\theta_{\text{res},7}^{\text{id}}$	$\theta_{\text{res},8}^{\text{id}}$	$\theta_{\text{res},9}^{\text{id}}$
-0.8 %	-18.4 %	-15.9 %	-21.6 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %
$\theta_{\text{res},10}^{\text{id}}$	$\theta_{\text{res},11}^{\text{id}}$	$\theta_{\text{res},12}^{\text{id}}$	$\theta_{\text{res},13}^{\text{id}}$	$\theta_{\text{res},14}^{\text{id}}$	$\theta_{\text{res},15}^{\text{id}}$	$\theta_{\text{res},16}^{\text{id}}$	$\theta_{\text{res},17}^{\text{id}}$	
0.0 %	0.0 %	0.0 %	-0.0 %	0.0 %	-0.0 %	0.0 %	0.0 %	

Tab. 5.2: Prozentuale Abweichungen der besten Näherung $\theta_{\text{res}}^{\text{id}}$ von den tatsächlichen Systemparametern im Fall $I_{\text{msr}} = \{5, 6, 8\}$

für $T_{\text{flw}}(t)$ zu kommen [29, Folien 7–12] und im besten Fall sogar gleichzeitig auch noch einen für $T_{\text{rtn}}(t)$ zu bestimmen [50, Folie 5]. Demzufolge erscheint mit Blick auf die Praxis bestenfalls die Annahme $I_{\text{msr}} = \{5, 6, 8\}$ als gerechtfertigt. Wenn für das Optimierungsproblem nur die Zustandsgrößen gemäß der zuletzt genannten Indexmenge als messbar angenommen werden, und die vorherigen Berechnungen unter sonst gleichen Bedingungen wiederholt werden, sehen die Resultate so aus wie in der Tabelle 5.2.

Die Resultate aus den beiden Tabellen sowie weitere Test mit anderen zufälligen Startwerten, die die gleichen, oben erwähnten Bedingungen erfüllen, zeigen, dass die Parameter des Systems sowohl unter der Voraussetzung $I_{\text{msr}} = I_{\text{all}}$ als auch unter der praxisrelevanten Annahme $I_{\text{msr}} = \{5, 6, 8\}$ prinzipiell identifizierbar sind. Für die Genauigkeit der Ergebnisse spielen die zufällig ausgetesteten Startwerte eine entscheidende Rolle. So können Ergebnisse wie aus Tabelle 5.1 auch im Fall $I_{\text{msr}} = \{5, 6, 8\}$ erreicht werden, aber umgekehrt kann selbst $I_{\text{msr}} = I_{\text{all}}$ auf ähnliche Werte führen wie in Tabelle 5.2, weil lokale Minima für unterschiedliche Konstellationen der durch θ_i^{id} für $i \in \{2, 3, 4\}$ repräsentierten Wärmekapazitäten C_{room} , C_{wall} , C_{fc} existieren. Für sehr günstige Startwerte kann in beiden Fällen die Abweichung sogar für $\theta_{\text{res},i}^{\text{id}}$ für $i \in \{2, 3, 4\}$ bei 0.0 %, während für sehr ungünstige das DAE-System nicht mehr gelöst werden kann und somit Ipopt gar keine Lösung liefert. Weil in der Praxis allerdings nicht die Möglichkeit besteht, die berechneten Ergebnisse mit den tatsächlichen zu vergleichen und weil, wie soeben gezeigt, nicht jede Initialisierung der Startwerte auf das globale Minimum führt, ist für die Entwicklung einer neuen Steuerstrategie ein modellbasierter Ansatz wenig empfehlenswert. Das trifft selbst dann zu, wenn garantiert werden könnte, dass stets recht gute Startwerte für die Optimierung auswählbar sind, denn es ist kaum zu erwarten, dass der Mikrocontroller einer Pumpe über hinreichenden Speicher und Prozessorleistung verfügt, um ein Optimierungsproblem mit 129 Variablen und 107 Nebenbedingungen mit einem komplexen Algorithmus wie Ipopt zu lösen. Die Schlussfolgerung, die sich draus ziehen lässt, lautet, dass für die Entwicklung von neuen Verfahren für die Regelung der Drehzahl einer Pumpe modellfreie Ansätze modellbasierten vorzuziehen sind.

6 Etablierte Steuerstrategien

Dieses Kapitel startet mit der Formulierung des Steuerproblems, bei dem die Drehzahl $n(t)$ der Heizungsumwälzpumpe so zu bestimmen ist, dass einerseits der Betrag des Integrals über die hydraulische Leistung möglichst klein wird, andererseits aber auch der Wunsch des Nutzers hinsichtlich eines genügend warmen Raumes Berücksichtigung findet. Mithilfe von theoretischen Überlegungen, die sich auf das Optimalitätsprinzip von Bellman stützen, können Referenzlösungen konstruiert werden, die der optimalen Lösung sehr nahekommen. Durch die Theorie entstehen so Vergleichs- und Bewertungskriterien für alle praxisrelevanten Lösungen. Außerdem werden unter dem Aspekt der praktischen Umsetzbarkeit, unter anderem durch Auswertung von Patenten, klassische Steuerstrategien präsentiert. Dabei erstreckt sich die Diskussion von elementaren, klassischen Ideen über den gegenwärtig besten Algorithmus von Grundfos bis hin zu einem neueren Ansatz von KSB.

6.1 Formulierung eines optimalen Steuerproblems

Mithilfe der im Abschnitt 5.1 eingeführten Notation lässt sich als Benchmark-Problem für die Suche nach der energetisch günstigsten Drehzahlsteuerung für die Umwälzpumpe in einem Einraum-Heizungssystem bei Sicherstellung einer ausreichenden Wärmezufuhr das optimale Steuerproblem mit der Zielfunktion J und der Nebenbedingung G formulieren [41, S. 13 f. (Problem 1.1.8)].

$$\begin{aligned} \dot{\mathbf{y}}(t) &= \mathbf{f}(t, \mathbf{y}(t), \mathbf{d}(t), u(t); \boldsymbol{\theta}_d) \quad \text{für } t \in [t_0, t_f], \quad \mathbf{y}(t_0) = \mathbf{y}_0 \\ J(\mathbf{y}(t), u(t)) &= \rho_{wg} \cdot \int_{t_0}^{t_f} H(t) \cdot \dot{V}(t) dt \rightarrow \min \\ G(\mathbf{y}(t), u(t)) &= \max_{t \in [t_0, t_f]} (T_{\text{set}}(t) - T_{\text{room}}(t)) - T_{\text{tol}} \leq 0 \end{aligned} \quad (50)$$

Hierbei ist die Solltemperatur $T_{\text{set}}(t)$ definiert als die Temperatur, die der Raum bei einer konstanten Drehzahlsteuerung $u(t) \equiv n_{\text{max}}$ mit der maximal möglichen Drehzahl annehmen würde. Dieser Definition liegt die Voraussetzung zugrunde, nach der die Pumpe so dimensioniert ist, dass ihr Betrieb mit maximaler Drehzahl das Heizungssystem zu keinem Zeitpunkt mit zu wenig Wärme versorgen würde und deshalb gewährleistet wäre, dass sich durch das Öffnen und Schließen des Thermostatventils, die vorgegebene Solltemperatur auch einstellen kann.

Des Weiteren wird für die Einhaltung der Nebenbedingung als Toleranzschwelle $T_{\text{tol}} = 0.45 \text{ K}$ festgelegt. Dieser Wert orientiert sich an den Untersuchungen von Battistel et al. [9, Fig. 3], nach denen ein Unterschied in dieser Höhe bei Umgebungstemperaturen um 24°C bereits von etwa 80 % aller Menschen wahrgenommen werden kann.

Ferner sind bei der Suche nach einer Steuerfunktion $u(t)$ die natürlichen Grenzen der kleinst- und größtmöglichen Drehzahl n_{min} und n_{max} , mit der die Pumpe betrieben werden kann, zu berücksichtigen. Aus diesem Grund sind als Steuerfunktionen nur Funktionen zulässig, welche die Bedingungen

$$\begin{aligned} u(t_0) &= u_0 := n_{\text{pump},0} \text{ und} \\ u(t) &\in [n_{\text{min}}, n_{\text{max}}] \text{ für alle } t \in [t_0, t_f] \end{aligned} \quad (51)$$

erfüllen.

6.2 Konstruktion einer Lösung nach Bellman

Vor der Diskussion von in der Praxis tatsächlich umsetzbaren Ansätzen, die in den weiteren Abschnitten dieses Kapitels präsentiert werden, soll es in diesem Abschnitt um die rein theoretische Konstruktion einer Lösung von (50) gehen, welche auf dem Optimalitätsprinzip von Bellman aufbaut. Die Kenntnis dieser Lösung ist insofern von Bedeutung, als dass sie als Referenzlösung zur Beurteilung der Güte unterschiedlicher Verfahren einsetzbar ist und somit insbesondere auch die Möglichkeit bietet, Vergleiche durchzuführen.

Ausgangspunkt für die Betrachtungen ist ein beliebiges $t_{\text{im}} \in]t_0, t_f[$, mit dem die Aufspaltung des Problems (50) in die zwei Probleme

$$\begin{aligned} \dot{\mathbf{y}}^{(A)}(t) &= \mathbf{f}(t, \mathbf{y}^{(A)}(t), \mathbf{d}(t), u^{(A)}(t); \boldsymbol{\theta}_d) \text{ für } t \in [t_0, t_{\text{im}}], \quad \mathbf{y}^{(A)}(t_0) = \mathbf{y}_0 \\ J^{(A)}(\mathbf{y}^{(A)}(t), u^{(A)}(t)) &= \rho_{\text{wg}} \cdot \int_{t_0}^{t_{\text{im}}} H(t) \cdot \dot{V}(t) dt \rightarrow \min \\ G^{(A)}(\mathbf{y}^{(A)}(t), u^{(A)}(t)) &= \max_{t \in [t_0, t_{\text{im}}]} (T_{\text{set}}(t) - T_{\text{room}}(t)) - T_{\text{tol}} \leq 0 \end{aligned} \quad (52)$$

und

$$\begin{aligned} \dot{\mathbf{y}}^{(B)}(t) &= \mathbf{f}(t, \mathbf{y}^{(B)}(t), \mathbf{d}(t), u^{(B)}(t); \boldsymbol{\theta}_d) \text{ für } t \in [t_{\text{im}}, t_f], \quad \mathbf{y}^{(B)}(t_0) = \mathbf{y}^{(A)}(t_{\text{im}}) \\ J^{(B)}(\mathbf{y}^{(B)}(t), u^{(B)}(t)) &= \rho_{\text{wg}} \cdot \int_{t_{\text{im}}}^{t_f} H(t) \cdot \dot{V}(t) dt \rightarrow \min \\ G^{(B)}(\mathbf{y}^{(B)}(t), u^{(B)}(t)) &= \max_{t \in [t_{\text{im}}, t_f]} (T_{\text{set}}(t) - T_{\text{room}}(t)) - T_{\text{tol}} \leq 0 \end{aligned} \quad (53)$$

vollzogen wird. Wenn nun $u_*^{(A)}$ eine optimale Steuerung von (52) auf $[t_0, t_{\text{im}}]$ ist

und analog $u_*^{(B)}$ eine von (53) auf $[t_{\text{im}}, t_f]$, so besagt das bellmansche Optimalitätsprinzip, dass die durch

$$u_*(t) := \begin{cases} u_*^{(A)}(t) & \text{für } t \in [t_0, t_{\text{im}}] \\ u_*^{(B)}(t) & \text{für } t \in [t_{\text{im}}, t_f] \end{cases}$$

definierte Steuerung das Optimierungsproblem (50) löst [16, S. 83 (§ 3.)]. Es ist nicht schwer zu erkennen, dass das Problem (52) nach dem gleichen Prinzip beliebig oft weiter aufteilbar ist [70, S. 55 (Abschn. 3.3)]. Für eine äquidistante Unterteilung des Intervalls $[t_0, t_f]$ in $M^{\text{Bm}} := (t_f - t_0) / \Delta t^{\text{Bm}}$ Teilintervalle mit den Stützpunkten $t_j^{\text{Bm}} := t_0 + j \cdot \Delta t^{\text{Bm}}$ sowie der Schrittweite Δt^{Bm} ist ergo eine optimale Steuerung von (50) konstruierbar, indem die M^{Bm} Teilprobleme

$$\begin{aligned} \dot{\mathbf{y}}^{(j)}(t) &= \mathbf{f}(t, \mathbf{y}^{(j)}(t), \mathbf{d}(t), u^{\text{Bm},(j)}(t); \boldsymbol{\theta}_d) \quad \text{für } t \in [t_{j-1}^{\text{Bm}}, t_j^{\text{Bm}}], \quad \mathbf{y}^{(j)}(t_{j-1}^{\text{Bm}}) = \mathbf{y}^{(j-1)}(t_{j-1}^{\text{Bm}}) \\ J^{(j)}(\mathbf{y}^{(j)}(t), u^{\text{Bm},(j)}(t)) &= \rho_{\text{wg}} g \cdot \int_{t_{j-1}^{\text{Bm}}}^{t_j^{\text{Bm}}} H(t) \cdot \dot{V}(t) dt \rightarrow \min \\ G^{(j)}(\mathbf{y}^{(j)}(t), u^{\text{Bm},(j)}(t)) &= \max_{t \in [t_{j-1}^{\text{Bm}}, t_j^{\text{Bm}}]} (T_{\text{set}}(t) - T_{\text{room}}(t)) - T_{\text{tot}} \leq 0 \end{aligned} \quad (54)$$

für $j \in \{1, \dots, M^{\text{Bm}}\}$ mit $\mathbf{y}^{(0)}(t_0) := \mathbf{y}_0$ sukzessive betrachtet werden. Für ein festes $j \in \{1, \dots, M^{\text{Bm}}\}$ ist die prinzipielle Idee dabei für $u^{\text{Bm},(j)}(t)$ ein Polynom ersten Grades der Form

$$\tilde{u}^{\text{Bm},(j)}(t) := u_{j-1} + \frac{t - t_{j-1}^{\text{Bm}}}{\Delta t^{\text{Bm}}} \cdot (u_j - u_{j-1}) \quad \text{für } t \in [t_{j-1}^{\text{Bm}}, t_j^{\text{Bm}}] \quad (55)$$

anzusetzen. Wegen des sukzessiven Vorgehens ist für $j > 1$ von den zwei Parametern u_{j-1} und u_j , welche diese lineare Funktion charakterisieren, der Wert von u_{j-1} stets bereits bekannt, wenn das $j-1$ -te Teilproblem zuvor gelöst wurde. Das gilt ebenso für den Fall $j = 1$, weil u_0 durch die Bedingung (51) für die Zulässigkeit einer Steuerfunktion vorgegeben ist. Daraus ergibt sich, dass während der Lösung des j -ten Teilproblems effektiv nur der Parameter u_j zu ermitteln ist. Um einen möglichst guten Wert für u_j zu finden, wird das Intervall $[n_{\text{min}}, n_{\text{max}}]$ der zulässigen Werte äquidistant mit einer Schrittweite Δn^{Bm} diskretisiert. Für jeden der Werte $n_i := n_{\text{min}} + i \cdot \Delta n^{\text{Bm}}$ mit $i \in \{1, \dots, N^{\text{Bm}}\}$ und $N^{\text{Bm}} := (n_{\text{max}} - n_{\text{min}}) / \Delta n^{\text{Bm}}$ wird dann das DGL-System aus (54) gelöst und anschließend die Nebenbedingung $G^{(j)}$ geprüft sowie der Wert der Zielfunktion $J^{(j)}$ berechnet. Dasjenige i_{opt} , bei dem einerseits die Nebenbedingung erfüllt bleibt und das andererseits bei diesem Teilproblem auf den kleinsten Wert für die Zielfunktion führt, wird danach für die Zuweisung $u_j := n_{i_{\text{opt}}}$ genutzt.

Nachdem auf diese Weise alle M^{Bm} Teilprobleme der Reihe nach behandelt

wurden, lautet die Vorschrift für die konstruierte Steuerfunktion

$$\tilde{u}_*^{\text{Bm}}: [t_0, t_f] \rightarrow [n_{\min}, n_{\max}]$$

also

$$\tilde{u}_*^{\text{Bm}}(t) := \tilde{u}^{\text{Bm},(j)}(t) \text{ für } t \in [t_{j-1}^{\text{Bm}}, t_j^{\text{Bm}}]. \quad (56)$$

Wie eine Abschätzung von Nishikawa [108, S. 3 (Gl. 7)] zeigt, stellt diese aus stückweise linearen Funktionen konstruierte Steuerfunktion eine sehr gute Näherung an die optimale Lösung von (50) dar, falls sowohl Δt^{Bm} als auch Δn^{Bm} hinreichend klein sind. Der Pseudocode 1 fasst das gesamte Vorgehen zur Bestimmung von \tilde{u}_*^{Bm} noch einmal übersichtlich zusammen.

Data: $t_0, t_f, \Delta t^{\text{Bm}}, n_{\min}, n_{\max}, \Delta n^{\text{Bm}}$

Result: Näherungslösung $\tilde{u}_*^{\text{Bm}}: [t_0, t_f] \rightarrow [n_{\min}, n_{\max}]$ für das Steuerproblem (50)

$M^{\text{Bm}} \leftarrow (t_f - t_0) / \Delta t^{\text{Bm}}$

$N^{\text{Bm}} \leftarrow (n_{\max} - n_{\min}) / \Delta n^{\text{Bm}}$

$u_0 \leftarrow \frac{1}{2} \cdot (n_{\min} + n_{\max})$

for $j \leftarrow 1$ **to** M^{Bm} **do**

$J_{\text{opt}}^{(j)} \leftarrow \infty$

$i_{\text{opt}} \leftarrow \infty$

for $i \leftarrow 1$ **to** N^{Bm} **do**

$n_i \leftarrow n_{\min} + i \cdot \Delta n^{\text{Bm}}$

$u_j \leftarrow n_i$

$\tilde{u}^{\text{Bm},(j)}(t) \leftarrow$ Definition nach Gleichung (55)

$\mathbf{y}^{(j)} \leftarrow$ Lösung des DGL-Systems aus Gleichung (54) mit $\tilde{u}^{\text{Bm},(j)}$ als Steuerfunktion

$G^{(j)}, J^{(j)} \leftarrow$ Berechnung gemäß Formeln aus (54) mithilfe von $\mathbf{y}^{(j)}$

if $G^{(j)} \leq 0$ **then**

if $J^{(j)} < J_{\text{opt}}^{(j)}$ **then**

$i_{\text{opt}} \leftarrow i$

$J_{\text{opt}}^{(j)} \leftarrow J^{(j)}$

end

end

end

$u_j \leftarrow n_{i_{\text{opt}}}$

$\tilde{u}^{\text{Bm},(j)}(t) \leftarrow$ Definition nach Gleichung (55)

end

$\tilde{u}_*^{\text{Bm}}(t) \leftarrow$ Konstruktion nach Vorschrift (56)

Alg. 1: Konstruktion einer optimalen Lösung nach Bellman

6.3 Elementare Steuerstrategien

Die im Folgenden präsentierten Steuerstrategien sind grundlegende Ansätze, um im Sinne der Problemstellung (50) in einem Heizsystem eine möglichst gute Drehzahl für die Pumpe zu wählen. Wie sich im Abschnitt 6.4 über erweiterte Steuerstrategien noch herausstellen wird, bilden diese elementaren Steuerstrategien häufig den Kern für viele der gegenwärtig genutzten oder sich noch in Entwicklung befindenden, aufwendigeren Strategien. In ihrer elementaren Reinform steht und fällt die Güte der Lösung mit den Einstellungen für die Werte der jeweils zugrunde liegenden Parameter.

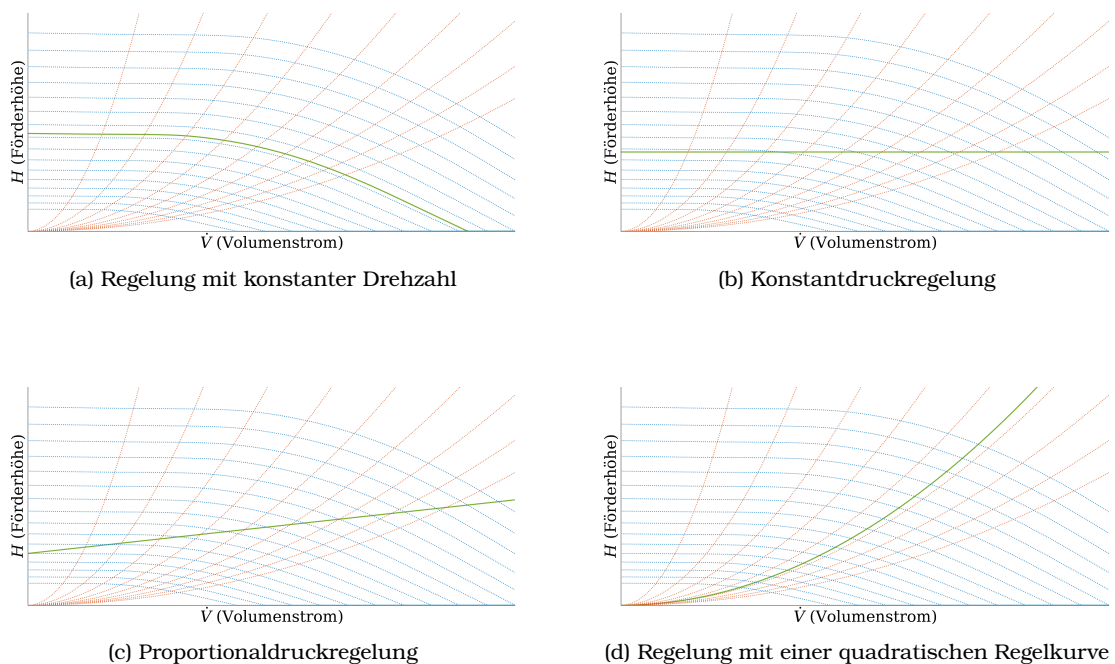


Abb. 6.1: Regelkurven (grün) der vier elementaren Steuerstrategien [116, S. 5 (Fig. 3)]

6.3.1 Regelung mit konstanter Drehzahl

Der simpelste Ansatz für das Finden einer möglichst guten Steuerung im Sinne des Problems (50) besteht darin, eine passende Drehzahl n^{cs} zu schätzen und die Pumpe dauerhaft mit dieser einen Drehzahl zu betreiben:

$$u^{\text{cs}}(t + \Delta t) := n^{\text{cs}}.$$

Das Einsparpotential hinsichtlich der hydraulischen Energie ist hier recht begrenzt, weil der Betriebspunkt der Pumpe nur auf einer Pumpenkennli-

nie (siehe Diagramm 6.1a) wandert und so die Förderhöhe mit abnehmenden Volumenstrom zunimmt [116, S. 3]. Aufgrund der rechtlichen Vorgaben der schon im Abschnitt 3.2 diskutierten Energiesparverordnung kann diese Lösung heutzutage ohnehin in den meisten Heizsystemen nicht mehr verwendet werden (vgl. Verordnung über energiesparenden Wärmeschutz und energiesparende Anlagentechnik bei Gebäuden in der Fassung der Bekanntmachung vom 24. Juli 2007 (BGBl. I S. 1519), die zuletzt durch die Zweite Verordnung zur Änderung der Energieeinsparverordnung vom 18. November 2013 (BGBl. I S. 3951) geändert worden ist, § 14 Absatz 3).

6.3.2 Konstantdruckregelung

Eine Konstantdruckregelung beruht auf der Festlegung eines konstanten Sollwertes für den gesamten Druckverlust der Anlage des betrachteten Heizsystems. Der gewünschte Druckverlust wird gemäß der Formel (18) durch die Ermittlung des entsprechenden Wertes H^{cp} für die Förderhöhe vorgegeben, wodurch sich eine Regelkurve der Form $H = H^{\text{cp}}$ (siehe Diagramm 6.1b) ergibt. Die Steuerung der Drehzahl erfolgt dabei nach

$$u^{\text{cp}}(t + \Delta t) := u^{\text{cp}}(t) \cdot \sqrt{\frac{H^{\text{cp}}}{H(t)}}.$$

6.3.3 Proportionaldruckregelung

Eine Proportionaldruckregelung besitzt eine lineare Regelkurve, welche sich mit den zwei einzustellenden Parametern m^{pp} und H^{pp} als $H = m^{\text{pp}} \cdot \dot{V} + H^{\text{pp}}$ (siehe Diagramm 6.1c) schreiben lässt. Für die Steuerung ergibt sich daraus

$$u^{\text{pp}}(t + \Delta t) := u^{\text{pp}}(t) \cdot \sqrt{\frac{m^{\text{pp}} \cdot \dot{V}(t) + H^{\text{pp}}}{H(t)}}.$$

Im Vergleich zu einer Konstantdruckregelung bietet eine solche, vom Volumenstrom abhängige Vorgabe der Förderhöhe bessere Chancen, das Integral über die hydraulische Leistung aus (50) zu reduzieren und somit mehr Energie einzusparen. Die Konstantdruckregelung stellt mathematisch gesehen nämlich den Spezialfall $m^{\text{pp}} = 0$ der Proportionaldruckregelung dar.

6.3.4 Regelung auf einer quadratischen Regelkurve

Die letzte elementare Regelung baut auf einer quadratischen Kurve der Gestalt $H = a^{\text{qp}} \cdot \dot{V}^2 + H^{\text{qp}}$ (siehe Diagramm 6.1d) auf, welche mithilfe der beiden

Parameter a^{qP} und H^{qP} definiert wird. Die Steuerung wird somit durch die Vorschrift

$$u^{\text{qP}}(t + \Delta t) := u^{\text{qP}}(t) \cdot \sqrt{\frac{a^{\text{qP}} \cdot \dot{V}(t)^2 + H^{\text{qP}}}{H(t)}}$$

beschrieben. Der natürliche Vorteil dieser Art der Regelkurve gegenüber einer linearen ist, dass sie dieselbe Gestalt wie eine Anlagenkennlinie aus Formel (22) aufweist, wodurch sie bei Wahl des selben Bezugspunktes unterhalb der linearen Kurve liegt.

6.4 Erweiterte Steuerstrategien

6.4.1 Autoadapt von Grundfos

In einem unveröffentlichten Paper [39] hat der Autor dieser Arbeit die Historie der Entwicklung erweiterter Steuerstrategien des Pumpenherstellers Grundfos detailliert zusammengefasst. An der Spitze dieser Entwicklung steht eine Steuerstrategie namens „Autoadapt“, die erstmals in die Software der Grundfos Alpha 2 Pumpe integriert wurde und die auch etliche Jahre nach ihrer Markteinführung noch zu den effektivsten Lösungen zählt [77], [123, S. 64 f.]. Der vorliegende Unterabschnitt widmet sich deswegen der Erläuterung des Autoadapt Algorithmus. Neben der vom Autor erarbeiteten Übersicht aus dem unveröffentlichten Paper wird dabei als Quelle im Wesentlichen auf die Inhalte aus dem zugehörigen Patent zurückgegriffen [46, S. 2–9 (Sp. 2–15)].

Obwohl die genauen Details des Autoadapt Algorithmus von Grundfos öffentlich an keiner Stelle bekanntgegeben wurden, bieten die vorhandenen, offiziellen Beschreibungen und Skizzen einige Anhaltspunkte dafür, wie die mathematische Formulierung dieser Steuerstrategie in etwa aussieht. Demnach ist davon auszugehen, dass eine Implementierung auf Basis der folgenden mathematischen Darstellung der tatsächlichen, unbekanntenen Implementierung vermutlich recht nahe kommt. Trotzdem soll hier nicht unerwähnt bleiben, dass bei einigen Erklärungen von Grundfos ein gewisser Interpretationsspielraum vorhanden ist und dass vom Autor dieser Arbeit an einigen Stellen deshalb die am plausibelsten erscheinenden Interpretationen ausgewählt werden mussten.

Die grundlegende Idee der Autoadapt Strategie besteht darin, die Drehzahl der Pumpe auf einer linearen oder quadratischen Regelkurve K_{mid} (siehe Abschnitte 6.3.3 und 6.3.4) zu steuern, welche mittig zu einer vorher ermittelten, minimalen Anlagenkennlinie K_{min} und maximalen Anlagenkennlinie K_{max} liegt. Während der Steuerung wird beobachtet, ob und auf welche Weise ein vorher festgelegter unterer Grenzpunkt $G_{\text{min}} = (\dot{V}_{G_{\text{min}}}, H_{G_{\text{min}}})$ oder oberer Grenzpunkt $G_{\text{max}} = (\dot{V}_{G_{\text{max}}}, H_{G_{\text{max}}})$ auf K_{mid} angefahren wird. Das Erreichen von G_{min} oder G_{max} führt zur Aktualisierung der Regelkurve inklusive neuer Grenzpunk-

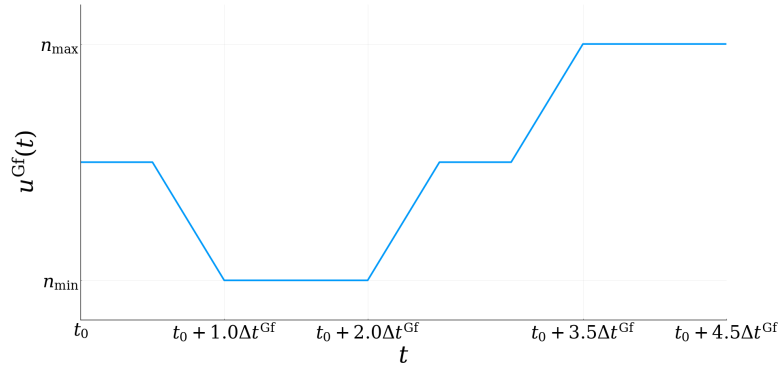


Abb. 6.2: Drehzahlsteuerung, die beim Autoadapt Algorithmus von Grundfos zur Initialisierung der minimalen und maximalen Anlagenkennlinien K_{\min} und K_{\max} verwendet wird

te. Zu diesem Zweck werden, abhängig davon, um welchen Grenzpunkt es sich handelt, durch gezieltes Anfahren einer besonders niedrigen oder hohen Drehzahl kurzzeitig für eine Dauer Δt^{Gf} bewusst eine Unter- oder Überversorgung im System erzeugt. Mithilfe der daraus gewonnenen charakteristischen Kennwerte werden anschließend K_{\min} oder K_{\max} und damit schließlich ebenso K_{mid} neu berechnet. Das Anpassen der Regelkurve K_{mid} soll dabei dafür sorgen, dass die Drehzahl der Pumpe stets im Einklang mit dem aktuellen Wärmebedarf der Anlage geregelt werden kann.

Die Initialisierung von K_{\min} und K_{\max} kann erst nach Ermittlung von vier Messwertepaaren $(\dot{V}(t_i), H(t_i))$ für $i \in \{1, 2, 3, 4\}$ erfolgen. Zu diesem Zweck werden die drei Drehzahlen n_{\min} , $\frac{1}{2} \cdot (n_{\min} + n_{\max})$ und n_{\max} mit stetigen Übergängen in Form von

$$u^{\text{Gf}}(t) := \begin{cases} \frac{1}{2} \cdot (n_{\min} + n_{\max}) & \text{für } t \in [t_0, t_0 + 0.5\Delta t^{\text{Gf}}] \\ -\frac{t-(t_0+\Delta t^{\text{Gf}})}{\Delta t^{\text{Gf}}} \cdot (n_{\max} - n_{\min}) + n_{\min} & \text{für } t \in [t_0 + 0.5\Delta t^{\text{Gf}}, t_0 + 1.0\Delta t^{\text{Gf}}] \\ n_{\min} & \text{für } t \in [t_0 + 1.0\Delta t^{\text{Gf}}, t_0 + 2.0\Delta t^{\text{Gf}}] \\ \frac{t-(t_0+2\Delta t^{\text{Gf}})}{\Delta t^{\text{Gf}}} \cdot (n_{\max} - n_{\min}) + n_{\min} & \text{für } t \in [t_0 + 2.0\Delta t^{\text{Gf}}, t_0 + 2.5\Delta t^{\text{Gf}}] \\ \frac{1}{2} \cdot (n_{\min} + n_{\max}) & \text{für } t \in [t_0 + 2.5\Delta t^{\text{Gf}}, t_0 + 3.0\Delta t^{\text{Gf}}] \\ \frac{t-(t_0+3.5\Delta t^{\text{Gf}})}{\Delta t^{\text{Gf}}} \cdot (n_{\max} - n_{\min}) + n_{\max} & \text{für } t \in [t_0 + 3.0\Delta t^{\text{Gf}}, t_0 + 3.5\Delta t^{\text{Gf}}] \\ n_{\max} & \text{für } t \in [t_0 + 3.5\Delta t^{\text{Gf}}, t_0 + 4.5\Delta t^{\text{Gf}}] \end{cases}$$

für eine Dauer von jeweils Δt^{Gf} kontrolliert angefahren. Der Graph aus dem Diagramm 6.2 veranschaulicht die Steuerung der Drehzahl während dieser Initialisierung. Bei den benötigten Messwerten handelt es sich um die Volumenströme $\dot{V}(t_i)$ sowie die entsprechenden Förderhöhen $H(t_i)$ zu den vier Zeitpunkten $t_1 = t_0 + \Delta t^{\text{Gf}}$, $t_2 = t_0 + 2.0\Delta t^{\text{Gf}}$, $t_3 = t_0 + 3.5\Delta t^{\text{Gf}}$ und $t_4 = t_0 + 4.5\Delta t^{\text{Gf}}$.

Aus ihnen ergeben sich die drei unbekannt Parameter a_{\min} , a_{\max} und H_0 der beiden Kurven $K_{\min}: H = a_{\min} \cdot \dot{V}^2 + H_0$ und $K_{\max}: H = a_{\max} \cdot \dot{V}^2 + H_0$, indem das Normalgleichungssystem

$$\mathcal{M}^T \mathcal{M} \mathbf{x} = \mathcal{M}^T \mathbf{b}$$

mit $\mathcal{M} = \begin{pmatrix} \dot{V}(t_1)^2 & 0 & 1 \\ \dot{V}(t_2)^2 & 0 & 1 \\ 0 & \dot{V}(t_3)^2 & 1 \\ 0 & \dot{V}(t_4)^2 & 1 \\ 0 & 0 & 1 \end{pmatrix}$, $\mathbf{b} = \begin{pmatrix} H(t_1) \\ H(t_2) \\ H(t_3) \\ H(t_4) \\ 0.0 \end{pmatrix}$ und $\mathbf{x} = \begin{pmatrix} a_{\min} \\ a_{\max} \\ H_0 \end{pmatrix}$ (57)

gelöst wird. Außerdem sorgen die zwei weiteren Messwertpaare $(\dot{V}(t_i), H(t_i))$ für $i \in \{5, 6\}$, die ebenfalls während der anfänglichen Steuerung zu den Zeitpunkten $t_5 = t_0 + 0.5\Delta t^{\text{Gf}}$ und $t_6 = t_0 + 3.0\Delta t^{\text{Gf}}$ aufzunehmen sind, für die Möglichkeit eine Initialisierung von $\dot{V}_{G_{\min}}$ und $\dot{V}_{G_{\max}}$ in der Gestalt

$$\dot{V}_{G_{\min}} := (1 - q^{\text{Gf}}) \cdot \frac{1}{2} \cdot (\dot{V}(t_5) + \dot{V}(t_6)) \quad \text{und} \quad \dot{V}_{G_{\max}} := (1 + q^{\text{Gf}}) \cdot \frac{1}{2} \cdot (\dot{V}(t_5) + \dot{V}(t_6)) \quad (58)$$

vorzunehmen.

Nach Bestimmung von K_{\min} und K_{\max} sowie $\dot{V}_{G_{\min}}$ und $\dot{V}_{G_{\max}}$ kann die anfängliche Regelkurve

$$K_{\text{mid}}: H = a_{\text{mid}} \cdot \dot{V}^2 + H_0 \quad (59)$$

durch Berechnung des noch fehlenden Parameters a_{mid} so festgelegt werden, dass sie etwa 65 % unterhalb von K_{\max} und etwa 35 % oberhalb von K_{\min} liegt. Die dem Patent entnommene und nachbearbeitete Skizze 6.3 gibt Aufschluss darüber, in welchem mathematisch geometrischen Sinne diese Lagebeschreibung zu verstehen ist. Es wird offenbar gefordert, dass die Normale zur Kurve K_{mid} , welche durch den Punkt $G_{\max} = (\dot{V}_{G_{\max}}, H_{G_{\max}})$ verläuft, die Kurven K_{\max} und K_{\min} in den Punkten $S_1 = (\dot{V}_{S_1}, H_{S_1})$ und $S_2 = (\dot{V}_{S_2}, H_{S_2})$ derart schneidet, dass für das Verhältnis der Längen der Strecken $d_1 := \overline{S_1 G_{\max}}$ und $d_2 := \overline{G_{\max} S_2}$ bezüglich der Skalierungen $\sigma_{\dot{V}}$ in \dot{V} -Richtung und σ_H in H -Richtung die Bedingung

$$\frac{d_1}{d_2} = \frac{0.65}{0.35} \quad (60)$$

gilt. Mithilfe der beiden weiteren Punkte $\tilde{S}_1 = (\dot{V}_{S_1}, H_{G_{\max}})$ und $\tilde{S}_2 = (\dot{V}_{S_1}, H_{G_{\max}})$ entstehen zwei ähnliche Dreiecke $\tilde{S}_1 G_{\max} S_1$ und $\tilde{S}_2 G_{\max} S_2$, sodass aus Relation (60) auch die Gleichungen

$$\frac{(\dot{V}_{G_{\max}} - \dot{V}_{S_1}) / \sigma_{\dot{V}}}{(\dot{V}_{S_2} - \dot{V}_{G_{\max}}) / \sigma_{\dot{V}}} = \frac{0.65}{0.35} \quad \text{und} \quad \frac{(H_{S_1} - H_{G_{\max}}) / \sigma_H}{(H_{G_{\max}} - H_{S_2}) / \sigma_H} = \frac{0.65}{0.35}$$

Bestimmungsgleichung der Form

$$a_{\text{mid}}^4 + \frac{2(a_2 - b_2)b_3 - a_2b_1^2}{(a_2 - b_2)^2} \cdot a_{\text{mid}}^3 + \frac{2(a_2 - b_2)a_3 - a_1b_1(a_2 + b_2) + b_3^2}{(a_2 - b_2)^2} \cdot a_{\text{mid}}^2 + \frac{a_1(b_1b_3 - a_1b_2) - a_3(b_1^2 - 2b_3)}{(a_2 - b_2)^2} \cdot a_{\text{mid}} + \frac{a_1(a_1b_3 - a_3b_1) + a_3^2}{(a_2 - b_2)^2} = 0 \quad (61)$$

mit

$$\begin{aligned} a_1 &= \frac{1}{2a_{\text{max}}} \cdot \frac{(\sigma_H / \sigma_{\dot{V}})^2}{\dot{V}_{G_{\text{max}}}} & b_1 &= \frac{2a_{\text{min}}}{0.65 \cdot a_{\text{max}} + 0.35 \cdot a_{\text{min}}} \cdot \dot{V}_{G_{\text{max}}} \\ a_2 &= \frac{1}{a_{\text{max}}} \cdot \dot{V}_{G_{\text{max}}}^2 & b_2 &= \frac{0.65}{0.35 \cdot (0.65 \cdot a_{\text{max}} + 0.35 \cdot a_{\text{min}})} \cdot \dot{V}_{G_{\text{max}}}^2 \\ a_3 &= \frac{1}{2a_{\text{max}}} \cdot \left(\frac{\sigma_H}{\sigma_{\dot{V}}}\right)^2 & b_3 &= \frac{a_{\text{min}}}{0.35 \cdot (0.65 \cdot a_{\text{max}} + 0.35 \cdot a_{\text{min}})} \cdot \dot{V}_{G_{\text{max}}}^2 \end{aligned}$$

für a_{mid} aufgestellt werden, sofern a_{min} , a_{max} sowie $\dot{V}_{G_{\text{max}}}$ bekannt sind, was aufgrund der Formeln (57) und (58) der Fall ist.

Mit der Bestimmung von a_{mid} ist die Initialisierung vollständig abgeschlossen, sodass die eigentliche Steuerung der Pumpe auf der Regelkurve K_{mid} gemäß

$$u^{\text{Gf}}(t + \Delta t) := u^{\text{Gf}}(t) \cdot \sqrt{\frac{a_{\text{mid}} \cdot \dot{V}(t)^2 + H_0}{H(t)}} \quad (62)$$

starten kann. Solange die Bedingung $\dot{V}_{G_{\text{min}}} < \dot{V}(t) < \dot{V}_{G_{\text{max}}}$ erfüllt bleibt, verändert sich auch K_{mid} nicht. Wenn allerdings zu einem Zeitpunkt $t = t_{\text{test}}$ der untere Grenzpunkt G_{min} bzw. der obere Grenzpunkt G_{max} erreicht wird, führt das dazu, dass die aktuell maximale Anlagenkennlinie K_{max} bzw. die aktuell minimale Anlagenkennlinie K_{min} durch einen Drehzahltest neu zu ermitteln ist. Im Fall $\dot{V}(t) \leq \dot{V}_{G_{\text{min}}}$ hat der Test die Gestalt

$$u^{\text{Gf}}(t) := \begin{cases} \frac{t-t_{\text{test}}}{\frac{1}{2}\Delta t^{\text{Gf}}} \cdot (n_{\text{max}} - n(t_{\text{test}})) + n(t_{\text{test}}) & \text{für } t \in [t_{\text{test}}, t_{\text{test}} + \frac{1}{2}\Delta t^{\text{Gf}}] \\ n_{\text{max}} & \text{für } t \in [t_{\text{test}} + \frac{1}{2}\Delta t^{\text{Gf}}, t_{\text{test}} + \frac{3}{2}\Delta t^{\text{Gf}}] \end{cases}$$

und die daraus resultierende Aktualisierung lautet

$$\begin{aligned} \dot{V}_{G_{\text{min}}} &\rightarrow \dot{V}_{G_{\text{min}}} - \Delta \dot{V}^{\text{Gf}} \\ \dot{V}_{G_{\text{max}}} &\rightarrow \dot{V}_{G_{\text{max}}} - \Delta \dot{V}^{\text{Gf}} \\ a_{\text{max}} &\rightarrow \max \left(\frac{H(t_{\text{test}} + \frac{3}{2}\Delta t^{\text{Gf}}) - H_0}{\dot{V}(t_{\text{test}} + \frac{3}{2}\Delta t^{\text{Gf}})^2}, q^{\text{Gf}} \cdot a_{\text{max}} + (1 - q^{\text{Gf}}) \cdot a_{\text{min}} \right) \\ a_{\text{mid}} &\rightarrow \text{Neuberechnung nach Gleichung (61)}. \end{aligned}$$

Nach dem stetigen Übergang

$$u^{\text{Gf}}(t) := -\frac{t - (t_{\text{test}} + \frac{3}{2}\Delta t^{\text{Gf}})}{\frac{1}{2}\Delta t^{\text{Gf}}} \times \left(n_{\text{max}} - n_{\text{max}} \cdot \sqrt{\frac{a_{\text{mid}} \cdot \dot{V}(t_{\text{test}} + \frac{3}{2}\Delta t^{\text{Gf}})^2 + H_0}{H(t_{\text{test}} + \frac{3}{2}\Delta t^{\text{Gf}})}} \right) + n_{\text{max}}$$

für $t \in [t_{\text{test}} + \frac{3}{2}\Delta t^{\text{Gf}}, t_{\text{test}} + 2\Delta t^{\text{Gf}}]$ wird die Pumpe schließlich wieder gemäß Formel (62) auf der Regelkurve K_{mid} betrieben, welche zuvor dem neu berechneten Wert von a_{mid} angepasst wurde.

Im Fall $\dot{V}(t) \geq \dot{V}_{G_{\text{max}}}$ weisen bei dem Drehzahltest

$$u^{\text{Gf}}(t) := \begin{cases} -\frac{t-t_{\text{test}}}{\frac{1}{2}\Delta t^{\text{Gf}}} \cdot (n(t_{\text{test}}) - n_{\text{min}}) + n(t_{\text{test}}) & \text{für } t \in [t_{\text{test}}, t_{\text{test}} + \frac{1}{2}\Delta t^{\text{Gf}}] \\ n_{\text{min}} & \text{für } t \in [t_{\text{test}} + \frac{1}{2}\Delta t^{\text{Gf}}, t_{\text{test}} + \frac{3}{2}\Delta t^{\text{Gf}}] \end{cases}$$

die Aktualisierung

$$\begin{aligned} \dot{V}_{G_{\text{min}}} &\rightarrow \dot{V}_{G_{\text{min}}} + \Delta \dot{V}^{\text{Gf}} \\ \dot{V}_{G_{\text{max}}} &\rightarrow \dot{V}_{G_{\text{max}}} + \Delta \dot{V}^{\text{Gf}} \\ a_{\text{min}} &\rightarrow \min \left(\frac{H(t_{\text{test}} + \frac{3}{2}\Delta t^{\text{Gf}}) - H_0}{\dot{V}(t_{\text{test}} + \frac{3}{2}\Delta t^{\text{Gf}})^2}, q^{\text{Gf}} \cdot a_{\text{min}} + (1 - q^{\text{Gf}}) \cdot a_{\text{max}} \right) \\ a_{\text{mid}} &\rightarrow \text{Neuberechnung nach Gleichung (61)} \end{aligned}$$

und der darauf folgende stetige Übergang

$$u^{\text{Gf}}(t) := \frac{t - (t_{\text{test}} + \frac{3}{2}\Delta t^{\text{Gf}})}{\frac{1}{2}\Delta t^{\text{Gf}}} \times \left(n_{\text{min}} \cdot \sqrt{\frac{a_{\text{mid}} \cdot \dot{V}(t_{\text{test}} + \frac{3}{2}\Delta t^{\text{Gf}})^2 + H_0}{H(t_{\text{test}} + \frac{3}{2}\Delta t^{\text{Gf}})}} - n_{\text{min}} \right) + n_{\text{min}}$$

für $t \in [t_{\text{test}} + \frac{3}{2}\Delta t^{\text{Gf}}, t_{\text{test}} + 2\Delta t^{\text{Gf}}]$ eine ähnliche Struktur auf. Die Aktualisierungen werden im Wesentlichen durch die zwei Parameter $\Delta \dot{V}^{\text{Gf}}$ und q^{Gf} des Algorithmus beeinflusst. Dabei gibt $\Delta \dot{V}^{\text{Gf}} > 0 \frac{\text{m}^3}{\text{s}}$ den Betrag der Verschiebung des unteren und des oberen Grenzpunktes für den Volumenstrom an. Der Parameter $q^{\text{Gf}} \in [0.25, 0.75]$ hingegen wird benutzt, um die Änderungen von a_{max} bzw. a_{min} durch eine Konvexkombination von a_{min} und a_{max} zu beschränken, mit dem Ziel dadurch auch auf den zu aktualisierenden Wert von a_{mid} einwirken und so im Endeffekt Änderungen der Lage der Regelkurve K_{mid} aus Formel (59) auf sinnvolle Weise ebenfalls beschränken zu können.

6.4.2 Dynamic Control von KSB

Während der Durchführung des SOPHi-Projektes hat Martin Eckl eine neue Steuerstrategie erfunden, die sein Arbeitgeber, der Pumpenhersteller KSB, mit Beendigung des Projektes als Patent angemeldet hat. Die Strategie von Eckl, welche von KSB den offiziellen Namen „Dynamic Control“ verliehen bekommen hat, wird im nun folgenden Teil dieses Unterabschnitts präsentiert, wobei der Schlussbericht des SOPHi-Projektes [32, S. 59–66] sowie das von KSB angemeldete Patent [80] als Informationsbasis für die Ausführungen dienen. Obwohl im Projekt ORSYGET, dem Nachfolgeprojekt von SOPHi, die Ansätze bereits weiterentwickelt wurden [33, S. 16–27], beschränkt sich die vorliegende Arbeit auf Dynamic Control. Diese Strategie hat es nämlich, im Gegensatz zu der erweiterten Idee, welche den Namen „Advanced Dynamic Control“ trägt, bereits zur Marktreife geschafft [33, S. 35], [81, Abschn. 3].

Ausgangspunkt für die Regelung von KSB ist eine der statischen Regelkurven aus dem vorherigen Abschnitt 6.3. Während im Laufe des SOPHi-Projektes auf die Konstantdruckregelung zurückgegriffen wurde, [31, S. 1 f.] wird laut Patent aktuell die Proportionaldruckregelung bevorzugt, da diese als ideal für die Umsetzung der Dynamic Control Strategie angesehen wird. Prinzipiell ist aber auch die Nutzung einer Regelung mit einer statischen Kurve, die eine quadratische oder sogar eine weitere, davon abweichende Form aufweist, denkbar. Der Grundgedanke des Algorithmus besteht darin, die Drehzahl der Pumpe zunächst gemäß der ausgewählten Regelkurve zu steuern und unterdessen fortlaufend zu prüfen, wie der Volumenstrom sich ändert. Wenn sich dieser mindestens für die Dauer von Δt^{KSB} nur innerhalb der Grenzen einer vorgegebenen Toleranz $\Delta \dot{V}^{\text{KSB}}$ aufhält, wird damit begonnen, die Drehzahl unter Berücksichtigung des Parameters n^{KSB} kontinuierlich abzusenken. Sobald sich für den Volumenstrom ein Wert verzeichnen lässt, der außerhalb der Grenzen liegt, fällt die Regelung zurück auf die statische Kurve und die gesamte Prozedur wiederholt sich von vorn.

Die konkrete mathematische Umsetzung der Strategie macht es erforderlich, einen sogenannten Resetpunkt in Form eines Zeitpunktes t_{rs} zusammen mit dem dazugehörigen Volumenstroms \dot{V}_{rs} zwischenzuspeichern. Das Paar $(t_{\text{rs}}, \dot{V}_{\text{rs}})$, das zu Beginn durch

$$t_{\text{rs}} = t_0 \quad \text{und} \quad \dot{V}_{\text{rs}} = \dot{V}(t_0)$$

initialisiert wird, beinhaltet stets die Information darüber, seit wann sich der Volumenstrom gegenwärtig in welchem Toleranzbereich aufhält. Im Verlauf der Regelung spielt demzufolge die Bedingung

$$|\dot{V}(t) - \dot{V}_{\text{rs}}| \leq \Delta \dot{V}^{\text{KSB}} \tag{63}$$

eine zentrale Rolle. Falls zu einem Zeitpunkt $t = t_{\text{rs,new}}$ die Ungleichung (63)

verletzt wird, kommt es zu einer Aktualisierung des Resetpunktes:

$$t_{rs} = t_{rs,new} \quad \text{und} \quad \dot{V}_{rs} = \dot{V}(t_{rs,new}).$$

Erst wenn der Volumenstrom im Rahmen der Toleranz $\Delta \dot{V}^{KSB}$ für eine gewisse Dauer $t_{low} := t - (t_{rs} + \Delta t^{KSB})$, die über die Zeitspanne Δt^{KSB} hinausgeht, unverändert bleibt, also die Bedingung (63) hinreichend lange keine Aktualisierung des Resetpunktes impliziert, setzt die Absenkdynamik von KSB ein. Infolgedessen wird die elementare Steuerung der Drehzahl gemäß

$$u^{KSB}(t) := u_{stat}^{KSB}(t) \quad \text{für} \quad t_{rs} \leq t < t_{rs} + \Delta t^{KSB}, \quad (64)$$

welche sich einzig aus der statischen Kurve ergibt, in solchen Fällen durch die Steuerung

$$u^{KSB}(t) := u_{stat}^{KSB}(t) - u_{dyn}^{KSB}(t_{low}) \quad \text{für} \quad t_{rs} + \Delta t^{KSB} \leq t < t_{rs,new} \quad (65)$$

ersetzt. Der dynamische Anteil $u_{dyn}^{KSB}(t_{low})$ repräsentiert die Reduzierung der Drehzahl bezogen auf die von der statischen Kurve ermittelte Drehzahl $u_{stat}^{KSB}(t)$. In der Implementierung von KSB, in der die Dynamic Control Strategie in signalorientierter Form mit Matlab Simulink umgesetzt wurde, handelt es sich bei der Funktion $u_{dyn}^{KSB}(t_{low})$ um das Ausgangssignal, das von der Übertragungsfunktion $G^{KSB}(s) = \frac{1}{s+1}$ aus dem Eingangssignal $n^{KSB} \cdot t_{low}$ generiert wird [30]. Aus der Definition $G^{KSB}(s) = \mathcal{L}\{u_{dyn}^{KSB}(t_{low})\} / \mathcal{L}\{n^{KSB} \cdot t_{low}\}$ der Übertragungsfunktion, in welcher $\mathcal{L}\{\cdot\}$ die Laplace-Transformation symbolisiert [119, S. 403], folgt wegen $\mathcal{L}\{n^{KSB} \cdot t_{low}\} = n^{KSB} \cdot \frac{1}{s^2}$ zunächst $\mathcal{L}\{u_{dyn}^{KSB}(t_{low})\} = n^{KSB} \cdot \frac{1}{s^2 \cdot (s+1)}$, woraus sich letztendlich durch Hinzuziehen einer Korrespondenz-Tabelle die gesuchte Funktion

$$u_{dyn}^{KSB}(t_{low}) := n^{KSB} \cdot (\exp(-t_{low}) + t_{low} - 1)$$

bestimmen lässt [119, S. 406 (Tab. A.1, Z. 3 u. Z. 18)]. Solange die Beziehung (63) ihre Gültigkeit nicht einbüßt, befolgt der Dynamic Control Algorithmus die Vorschrift (65). Erst wenn die Notwendigkeit besteht, einen neuen Resetpunkt (t_{rs}, \dot{V}_{rs}) zu definieren, weil die Beziehung (63) nicht mehr erfüllt ist, fällt der Algorithmus wieder auf die rein elementare Regelung (64) zurück.

7 KI-Steuerstrategie

Nachdem im vorangegangenen Kapitel existierende Steuerstrategien zur Lösung von (50) präsentiert wurden, welche den strengen Anforderungen der Praxis genügen, widmet sich das aktuelle Kapitel einem neuen Verfahren, welches dieselben Anforderungen erfüllt und das der Autor im Laufe des SOPHi-Projektes konzipiert und danach noch weiter verfeinert hat. Es beruht auf dem Einsatz von KI-Algorithmen, deren Parametrierung auf Basis einer Testphase erfolgt, welche der eigentlichen Steuerphase zunächst vorausgeht.

Mit den Ergebnissen aus Kapitel 5 wurde die Erkenntnis gewonnen, dass auf Basis sämtlicher bestenfalls in der Praxis zugänglicher Messwerte die unbekannt Systemparameter nur mit unverhältnismäßig hoher Rechenkapazität und großem Zweitaufwand hinreichend genau identifizierbar sind. Die Konsequenz, die sich daraus ergibt, lautet, dass eine neue praxistaugliche Steuerstrategie entweder lediglich auf einem stark vereinfachten Modell basieren kann oder dass ihr so wie bei allen bekannten Steuerstrategien ein modellfreier Ansatz zugrunde liegt. Bei einem modellfreien Ansatz beschränkt sich eine mögliche Untersuchung des aktuellen Systemzustandes anhand aller verfügbaren Messwerte dann aber darauf, dass allenfalls eine grobe Charakterisierung des Zustands vorgenommen werden kann. Das hier vorgestellte, neue Verfahren, das im Wesentlichen auf einem Reinforcement Learning Ansatz in Verbindung mit einem neuronalen Netz aufbaut, wird jedoch zeigen, dass diese Restriktion für den Entwurf einer guten Pumpensteuerung kein Hindernis darstellt. Denn solange zumindest die Möglichkeit existiert, Zustände hinsichtlich des Aspekts der Unter- und Überversorgung des Systems zu bewerten, lässt sich die Drehzahl der Pumpe im Sinne der Problemstellung (50) steuern.

Um während der eigentlichen Steuerphase die Bewertung von Systemzuständen möglichst zuverlässig vornehmen zu können, bietet es sich an vorab eine Testphase durchzuführen. Während dieser anfänglichen Testphase werden gezielt fest vorgegebene Drehzahlen angefahren, wodurch viele verschiedene Systemzustände bewusst erzeugt werden sollen. Ebenso werden bei potenziell ähnlichen Zuständen eine Reihe unterschiedlicher Drehzahländerungen getestet und die sich jeweils einstellenden neuen Zustände beobachtet. Die während der einzelnen Tests aufgenommenen Messwerte werden zu neuen Größen weiterverarbeitet und abgespeichert. Am Ende der Testphase werden alle gespeicherten Werte schließlich verwendet, um die Parameter des neuronalen Netzes einer Action-Value-Funktion zu initialisieren. Diese Funktion wiederum spielt die entscheidende Rolle in der eigentlichen Steuerphase, in

der ein Reinforcement Learning Ansatz zum Tragen kommt, weil mit ihrer Hilfe für jeden Systemzustand eine Drehzahländerung ermittelt werden kann, die mit Bezug auf die Zielfunktion und auf die Nebenbedingung aus (50) günstig ist.

Dieses Kapitel startet mit einer kurzen Einführung in die für diese Arbeit wichtigen Aspekte des Reinforcement Learnings, bei der der Leser insbesondere mit der dazugehörigen Notation vertraut gemacht wird. Im darauf folgenden Abschnitt wird erklärt, wie die zweiteilige Testphase aufgebaut ist und welche Größen aus den Messwerten berechnet werden, die später bei der Initialisierung der Steuerphase eine Rolle spielen. Die Initialisierung wiederum wird im dritten Abschnitt genau beschrieben, bevor im letzten Abschnitt schließlich das Vorgehen der eigentlichen Steuerphase im Fokus steht.

7.1 Grundlagen des Reinforcement Learnings

Bei dem neuen, in den folgenden Abschnitten präsentierten Verfahren wird hauptsächlich auf Methoden aus dem Bereich des Reinforcement Learnings (Bestärkendes Lernen) zurückgegriffen. Obwohl dieses Forschungsgebiet bereits seit Ende der 1970er Jahre existiert [124, S. xvii] und obwohl solide mathematische Grundlagen schon seit Ende der 1990er Jahre bekannt sind, ist dessen Popularität in Bezug auf praktische Anwendungen erst in den letzten Jahren im Zuge der Verfügbarkeit höherer Rechenleistung gestiegen [124, S. xiii], [142, 0:00–0:42 min u. 3:16–3:50 min]. Der Einsatz von Methoden des Reinforcement Learnings ist für das gegebene Problem, bei dem eine optimale Steuerung von (50) gesucht wird, aus zwei Gründen günstig. Erstens gibt es in diesem Forschungsbereich Algorithmen, bei denen nicht vorausgesetzt wird, dass das dem Problem zugrundeliegende Modell bekannt ist. Das ist wichtig, weil nach den empirischen Untersuchungen aus Kapitel 5 davon ausgegangen werden sollte, dass nur modellfreien Verfahren auch praktische Bedeutung zukommen kann. Zweitens korrespondiert die Grundidee des Reinforcement Learnings mit den Umständen eines Heizsystems, welches mithilfe einer Pumpe zu steuern ist: Es ist a priori unklar, wie eine optimale Steuerung im Detail aussieht, aber es besteht die Möglichkeit etwas darüber auszusagen, ob getroffene Steuerentscheidungen eher gut oder eher nicht so gut waren, wodurch sich Informationen für zukünftige Entscheidungen sammeln lassen [142, 1:14–1:47 min]. Die Notation und die mathematischen Definitionen, die in diesem Abschnitt eingeführt und benutzt werden, sind eng angelehnt an die des Standardwerkes von Sutton und Barto [124, Kap. 3 u. Abschn. 10.3].

7.1.1 Definition eines Markow-Entscheidungsprozesses

Der erste Schritt für die Entwicklung eines Verfahrens, das im Kern auf den Ideen aus dem Bereich des Reinforcement Learnings aufbaut, ist stets die Formulierung eines Markow-Entscheidungsprozesses (MEP) [141, 1:29–1:48 min]. Ein MEP beschreibt die Dynamik eines Systems, in dem hier behandelten Fall, die des Benchmark-Systems, auf Basis einer Menge \mathcal{S} aller möglichen Zustände, die das System annehmen kann, sowie auf Basis der Mengen $\mathcal{A}(s)$ aller möglichen Aktionen, die im Zustand $s \in \mathcal{S}$ ausführbar sind. Wenn das System, welches im Kontext des Reinforcement Learnings auch als Umgebung bezeichnet wird, in einem Zustand $s \in \mathcal{S}$ vorliegt, trifft ein Entscheidungsträger, der sogenannte Agent, welcher hier durch die Pumpe verkörpert wird, die Wahl für eine Aktion $a \in \mathcal{A}(s)$. Nach Ausführung der Aktion a durch den Agenten ändert sich der Zustand $s \in \mathcal{S}$ der Umgebung in einen neuen Zustand $s' \in \mathcal{S}$ und der Agent erhält von der Umgebung eine Belohnung $r \in \mathcal{R} \subseteq \mathbb{R}$. Die Herausforderung für den Agenten besteht dabei darin, seine Entscheidungen so zu fällen, dass mittel- bis langfristig so viele Belohnungen wie nur eben möglich kassiert werden. Der gesamte Markow-Entscheidungsprozess ist durch eine Folge von abwechselnd aufeinander folgenden Zuständen $S_\tau \in \mathcal{S}$, Aktionen $A_\tau \in \mathcal{A}$ und Belohnungen $R_\tau \in \mathcal{R}$ mit $\tau \in \mathbb{N}$ in der Form

$$S_0, A_0, R_1, S_1, A_1, \dots, R_\tau, S_\tau, A_\tau, R_{\tau+1}, \dots$$

abbildbar. Unter der Annahme, dass \mathcal{S} , \mathcal{A} , und \mathcal{R} beschränkte Mengen sind und unter der Annahme, dass die Markov-Eigenschaft erfüllt ist, welche besagt, dass mit Kenntnis des aktuellen Zustandes alle Informationen über die vorausgegangene Interaktion zwischen Agent und Umgebung zur Verfügung stehen, die Einfluss auf den weiteren Verlauf des Prozesses haben, stellen $R_{\tau+1}$ und $S_{\tau+1}$ Zufallsvariablen mit diskreten Wahrscheinlichkeitsverteilungen dar, die nur von S_τ und A_τ abhängen. Die Dynamik des MEPs lässt sich deswegen formal durch eine Funktion $p: \mathcal{S} \times \mathcal{R} \times \mathcal{A} \times \mathcal{S} \rightarrow [0, 1]$ charakterisieren, bei der $p(s', r | s, a)$ die Wahrscheinlichkeit angibt, dass der Zustand der Umgebung sich zu $S_{\tau+1} = s'$ ändert und der Agent die Belohnung $R_{\tau+1} = r$ bekommt, falls sich die Umgebung zuvor im Zustand $S_\tau = s$ befunden und der Agent sich für die Aktion $A_\tau = a$ entschieden hat. Damit diese Interpretation der Funktionswerte von p im Sinne der Wahrscheinlichkeitstheorie konsistent ist, muss die Bedingung

$$\sum_{s' \in \mathcal{S}} \sum_{r \in \mathcal{R}} p(s', r | s, a) = 1 \quad \text{für alle } s \in \mathcal{S} \text{ und } a \in \mathcal{A}$$

gefordert werden. Sie stellt sicher, dass die Abbildung $p(\cdot, \cdot | s, a): \mathcal{S} \times \mathcal{R} \rightarrow [0, 1]$ für jede beliebige Wahl von $s \in \mathcal{S}$ und $a \in \mathcal{A}$ eine Wahrscheinlichkeitsverteilung wird. Eine weitere wichtige Annahme, die typischerweise getroffen wird und

die als Kommunikationsannahme bekannt ist, besagt, dass der Agent für jedes beliebige Paar s und s' von Zuständen stets die Möglichkeit hat Aktionen so auszuwählen, dass ein System, das sich im Zustand s befindet, nach Ausführung einer endlichen Anzahl von Aktionen in den Zustand s' übergeht [133, S. 10654 (Abschn. 2)].

7.1.2 Formale Lösung des Markow-Entscheidungsprozesses

Im vorherigen Unterabschnitt wurde bisher nur qualitativ erwähnt, dass der Agent bei der Wahl seiner Entscheidungen das Ziel verfolgt, die Belohnungen, die er mittel- bis langfristig kumuliert, zu maximieren. Die formale Quantifizierung dessen ist Gegenstand dieses Unterabschnitts. Dazu wird als Erstes der zukünftige Gewinn G_τ eingeführt. Bei einem Steuerprozess, wie er hier vorliegt, handelt es sich üblicherweise um eine fortlaufende Kette von Interaktionen zwischen Agent und Umgebung, was auf die Definition

$$G_\tau := R_{\tau+1} + \gamma R_{\tau+2} + \gamma^2 R_{\tau+3} + \dots = \sum_{\kappa=0}^{\infty} \gamma^\kappa R_{\tau+\kappa+1}. \quad (66)$$

führt. Der zukünftige Gewinn ist demnach eine mit einem Diskontierungsfaktor $\gamma \in [0, 1[$ gewichtete Summe aller noch ausstehenden Belohnungen. Wegen der Konvergenz der geometrischen Reihe ist sichergestellt, dass $G_\tau < \infty$ ist, wenn die Menge \mathcal{R} beschränkt bleibt, was nach Voraussetzung zutrifft. Mit dem Wert von γ ist kontrollierbar, wie groß der Einfluss in naher Zukunft auftretender Belohnungen ist. Für kleine γ nah bei Null liegt der Fokus stärker auf den Belohnungen, die für die nächsten Aktionen vergeben werden. Im Extremfall $\gamma = 0$ ist wegen $G_\tau = R_{\tau+1}$ sogar nur die unmittelbar folgende Belohnung relevant. Je näher sich der Wert von γ in Richtung Eins bewegt, desto wichtiger wird es für den Agenten langfristig zu planen, wenn es um die Maximierung des zukünftigen Gewinns geht, da die Bedeutung von weiter in der Zukunft liegenden Belohnungen zunimmt. Eine allgemein nützliche Relation, die diesen Umstand auf andere Weise veranschaulicht, ist durch die Rekursion

$$G_\tau = R_{\tau+1} + \gamma \cdot G_{\tau+1}$$

gegeben, welche leicht aus der Gleichung (66) herleitbar ist.

Das exakte Ziel des Agenten besteht nun darin, seine Entscheidungen so zu wählen, dass der Erwartungswert des zukünftigen Gewinns zu jedem beliebigen Zeitschritt τ maximal wird:

$$\mathbb{E}(G_\tau) = \mathbb{E}\left(\sum_{\kappa=0}^{\infty} \gamma^\kappa R_{\tau+\kappa+1}\right) \rightarrow \max. \quad (67)$$

Eine formale Lösung des oben stehenden Maximierungsproblems lässt sich

angeben, indem ein Bewertungsschema für Zustände und Aktionen erschaffen wird, welches alle denkbaren Handlungsstränge miteinbezieht, denen der Agent durch die Ausführung seiner künftigen Aktionen folgen kann. Ein Handlungsstrang π ist dabei mathematisch gesehen eine Abbildung $\mathcal{A} \times \mathcal{S} \rightarrow [0, 1]$ und wird auch Policy oder genauer gesagt stochastische Policy genannt. Ein Agent hält sich im Zeitschritt τ an die Vorgabe einer stochastischen Policy π , wenn er sich mit der Wahrscheinlichkeit $\pi(a | s)$ für die Aktion $A_\tau = a$ entscheidet, falls das System im Zustand $S_\tau = s$ vorliegt. Die Funktion $\pi(\cdot | s)$ soll demnach für jedes $s \in \mathcal{S}$ eine Wahrscheinlichkeitsverteilung auf der Menge $\mathcal{A}(s)$ sein, was die Forderung nach der Einhaltung der Bedingung

$$\sum_{a \in \mathcal{A}(s)} \pi(a | s) = 1 \quad \text{für alle } s \in \mathcal{S}$$

impliziert [138, 1:51–2:30 min]. Stochastische Policies, bei denen es für jeden Zustand $s \in \mathcal{S}$ genau eine Aktion $a \in \mathcal{A}(s)$ gibt, für welche die Gleichung $\pi(a | s) = 1$ gilt, stellen Spezialfälle dar und werden als deterministische Policies bezeichnet. Der Name rührt daher, dass die Aktionen eines Agenten, der sich an die Vorgabe einer deterministischen Policy hält, für jeden Zustand klar bestimmt sind. Deshalb ist es von der Notation her auch günstig, eine deterministische Policy als eine Abbildung von \mathcal{S} nach \mathcal{A} anzusehen und $\pi(s) = a$ zu schreiben.

Eine deterministische Policy π_* zu finden, die im Sinne des Problems (67) optimal ist, stellt ergo die prinzipielle Aufgabe für den Agenten dar. Zur Bewertung von Zuständen bezüglich einer Policy π dient hierbei die Value-Funktion

$$v_\pi(s) := \mathbb{E}_\pi(G_\tau | S_\tau = s) = \mathbb{E}_\pi\left(\sum_{\kappa=0}^{\infty} \gamma^\kappa R_{\tau+\kappa+1} | S_\tau = s\right). \quad (68)$$

Jene liefert auf Basis des zu erwartenden zukünftigen Gewinns einen Wert dafür, wie günstig es ist, wenn sich die Umgebung zu einem beliebigen Zeitschritt τ im Zustand S_τ befindet und der Agent sich für alle kommenden Entscheidungen an die Vorgabe der Policy π hält, was durch den Erwartungswert $\mathbb{E}_\pi(\cdot)$ symbolisiert wird. Überdies gibt es noch die Action-Value-Funktion

$$q_\pi(s, a) := \mathbb{E}_\pi(G_\tau | S_\tau = s, A_\tau = a) = \mathbb{E}_\pi\left(\sum_{\kappa=0}^{\infty} \gamma^\kappa R_{\tau+\kappa+1} | S_\tau = s, A_\tau = a\right), \quad (69)$$

die sich von der Value-Funktion insofern unterscheidet, als dass sie bei der Bewertung nicht nur den Zustand $S_\tau = s$, sondern auch die Aktion $A_\tau = a$, die der Agent im Zeitschritt τ ausführt, berücksichtigt.

Für fortdauernde Prozesse gibt es neben der wie bis hierhin vorgestellten formalen mathematischen Gestaltung eines MPEs, bei der der zukünftige Gewinn G_τ gemäß Gleichung (66) über eine Summe diskontierter Belohnungen

ausgedrückt wird, noch eine weitere alternative Formulierung, bei der mathematisch mit einer Durchschnittsbelohnung gearbeitet wird. Diese Alternative bietet sich speziell für fortdauernde Prozesse an, bei denen kurzfristige Belohnungen als gleichwertig zu langfristigen Belohnungen zu betrachten sind. Da dies im Fall der Steuerung einer Pumpe zutrifft und da für fortdauernde Prozesse die Formulierung mithilfe einer Durchschnittsbelohnung ohnehin unter den allermeisten Voraussetzungen empfehlenswert erscheint [124, S. 253 (Abschn. 10.4)], wird nun kurz aufgezeigt, zu welchen Veränderungen die Einführung einer Durchschnittsbelohnung führt. Die Durchschnittsbelohnung bezüglich einer Policy π wird definiert durch

$$\bar{R}(\pi) := \lim_{h \rightarrow \infty} \frac{1}{h} \sum_{\tau=1}^h \mathbb{E}_{\pi}(R_{\tau} | S_0, A_{0:\tau-1}) = \lim_{\tau \rightarrow \infty} \mathbb{E}_{\pi}(R_{\tau} | S_0, A_{0:\tau-1}) \quad (70)$$

und gibt die zu erwartende nächste Belohnung an, die der Agent erhält, wenn er ausgehend vom Anfangszustand S_0 hinreichend lange die durch π vorgegebenen Aktionen ausführt. Die am Ende vom Unterabschnitt 7.1.1 erläuterte Kommunikationsannahme stellt dabei sicher, dass der Limes existiert [133, S. 10654 (Abschn. 2)]. Der zukünftige diskontierte Gewinn aus Formel (66) wird in dieser neuen Formulierung durch den differentiellen Gewinn

$$G_{\tau} := R_{\tau+1} - \bar{R}(\pi) + R_{\tau+2} - \bar{R}(\pi) + R_{\tau+3} - \bar{R}(\pi) + \dots = \sum_{\kappa=0}^{\infty} (R_{\tau+\kappa+1} - \bar{R}(\pi))$$

ersetzt, sodass das Maximierungsproblem (67) die Gestalt

$$\mathbb{E}_{\pi}(G_{\tau}) = \mathbb{E}_{\pi}\left(\sum_{\kappa=0}^{\infty} (R_{\tau+\kappa+1} - \bar{R}(\pi))\right) \rightarrow \max \quad (71)$$

annimmt. In analoger Weise verändern sich die Value-Funktion und die Action-Value-Funktion aus den Gleichungen (68) und (69) zu

$$v_{\pi}(s) = \mathbb{E}_{\pi}\left(\sum_{\kappa=0}^{\infty} (R_{\tau+\kappa+1} - \bar{R}(\pi)) \mid S_{\tau} = s\right) \text{ und} \quad (72)$$

$$q_{\pi}(s, a) = \mathbb{E}_{\pi}\left(\sum_{\kappa=0}^{\infty} (R_{\tau+\kappa+1} - \bar{R}(\pi)) \mid S_{\tau} = s, A_{\tau} = a\right). \quad (73)$$

Die Value-Funktion erlaubt es mithilfe der partiellen Ordnung

$$\pi \leq \tilde{\pi} \Leftrightarrow v_{\pi}(s) \leq v_{\tilde{\pi}}(s) \quad \text{für alle } s \in \mathcal{S}$$

Vergleiche auf der Menge aller Policies anzustellen. Dabei existiert stets mindestens eine optimale, deterministische Policy π_{*} , die besser oder wenigstens

genauso gut wie alle anderen Policies ist, das heißt

$$\pi_* \geq \pi \quad \text{für jede Policy } \pi \quad (74)$$

[139, 2:26–3:30 min]. Alle optimalen Policies besitzen dieselbe optimale Value-Funktion

$$v_*(s) := \max_{\pi} (v_{\pi}(s)) \quad \text{für alle } s \in \mathcal{S}$$

sowie dieselbe optimale Action-Value-Funktion

$$q_*(s, a) := \max_{\pi} (q_{\pi}(s, a)) \quad \text{für alle } s \in \mathcal{S} \text{ und } a \in \mathcal{A}(s), \quad (75)$$

welche beide eindeutig bestimmt sind. Unter der Voraussetzung, dass letztere Funktion bekannt ist, lautet die rein formale Lösung des Maximierungsproblems (67) beziehungsweise (71) gerade

$$\pi_*(s) := \arg \max_{a \in \mathcal{A}(s)} q_*(s, a) \quad (76)$$

[140, 7:28–7:58 min]. Viele Algorithmen im Bereich des Reinforcement Learning bauen deshalb auf der Idee auf, eine Näherung für $q_*(s, a)$ zu suchen.

7.1.3 Gestaltung des Markow-Entscheidungsprozesses für das Benchmark-System

Abhängig von der konkreten Problemstellung sind oft eine Reihe unterschiedlicher Darstellungen für die Mengen der möglichen Zustände \mathcal{S} der Umgebung und der Aktionen $\mathcal{A}(s)$, die der Agent in einem Zustand s ausführen kann, denkbar. Gleiches trifft auf die Menge der Belohnungen \mathcal{R} zu. Obwohl diese Designfragen einen erheblichen Einfluss auf die Performance eines MEPs haben können, ist es nicht leicht, genaue, objektive Kriterien für ein vorteilhaftes Design anzugeben, sodass häufig auf Erfahrungswerte zurückgegriffen oder die Methode des systematischen Ausprobierens angewendet werden muss [124, S. 50 f.].

Die mathematische Beschreibung des MEPs des Benchmark-Systems startet zunächst mit der Vorgabe einer konstanten Zeitspanne Δt^{AI} , welche angibt, wie viel Zeit von einer Interaktion zwischen Agent und Umgebung bis zur nächsten vergeht. Das bedeutet, dass wenn t_0^{AI} den Zeitpunkt bezeichnet, an dem die Steuerphase beginnt, $t_{\tau}^{\text{AI}} := t_0^{\text{AI}} + \tau \cdot \Delta t^{\text{AI}}$ für den Zeitpunkt steht, an dem die Umgebung sich im Zustand S_{τ} befindet und der Agent die Aktion A_{τ} vornimmt, für die er die Belohnung $R_{\tau+1}$ kassiert.

Für die konkrete Definition der Menge aller möglichen Zustände \mathcal{S} drängt es sich in fast natürlicher Weise auf, einen Bezug zu dem Betriebspunkt der Pumpe herzustellen, sodass die Elemente der Menge \mathcal{S} als Vektoren mit zwei

Komponenten dargestellt werden. Weil es für die späteren Berechnungen einen Vorteil mit sich bringt, wird bei der Definition auch darauf geachtet, dass jede Vektorkomponente aus dem Intervall $[-1, 1]$ stammt. Für den Zustand $S_\tau \in \mathcal{S}$ zum Zeitpunkt t_τ sollen die beiden Vektorkomponenten den Betriebspunkt $(\dot{V}(t_\tau), H(t_\tau))$ repräsentieren. Weil dieser sich wie im Abschnitt 3.3 erläutert und wie in Abbildung 3.2 illustriert aus dem Schnittpunkt der Pumpenkennlinie mit der Anlagenkennlinie ergibt, können anstelle des Volumenstroms $\dot{V}(t_\tau)$ und der Förderhöhe $H(t_\tau)$ auch die Drehzahl $n_{\text{pump}}(t_\tau)$ der Pumpenkennlinie sowie der Parabelparameter $a_{\text{sys}}(t_\tau)$ der Anlagenkennlinie aus Gleichung (22) mit der Näherung $H_{\text{sys},0} = 0 \text{ m}$ als Grundlage für die Berechnung von $(S_\tau)_1$ und $(S_\tau)_2$ dienen. Für die erste Komponente wird die Drehzahl $n_{\text{pump}}(t_\tau) \in [n_{\text{min}}, n_{\text{max}}]$ in ganz einfacher Weise bijektiv auf das Intervall $[-1, 1]$ transformiert, sodass sich

$$(S_\tau)_1 := 2 \cdot \frac{n_{\text{pump}}(t_\tau) - n_{\text{min}}}{n_{\text{max}} - n_{\text{min}}} - 1 \quad (77)$$

ergibt. Hinsichtlich der zweiten Komponente ist das Vorgehen jedoch komplexer, weil eine lineare Transformation auf das Intervall $[-1, 1]$ in Analogie zur Drehzahl dazu führt, dass der Zusammenhang zwischen dem Parabelparameter und dem Ventilöffnungsgrad keine Berücksichtigung im Ergebnis findet, der Ventilöffnungsgrad jedoch intuitiv nützlicher erscheint, was die noch ausstehende und weiter unten folgende mathematische Formulierung der Belohnungen angeht. Daher wird der Parabelparameter $a_{\text{sys}}(t_\tau)$ nichtlinear auf das Intervall $[-1, 1]$ abgebildet, und zwar derart, dass der Bezug zum Ventilöffnungsgrad $\alpha(t_\tau)$ eine entscheidende Rolle spielt. Ausgehend von der maximalen Anlagenkennlinie

$$H = a_{\text{max}} \cdot \dot{V}^2, \quad (78)$$

welche sich bei einem weitestgehend geschlossenem Ventil einstellt, sowie der minimalen Anlagenkennlinie

$$H = a_{\text{min}} \cdot \dot{V}^2, \quad (79)$$

welche sich bei einem weitestgehend geöffnetem Ventil einstellt, seien die jeweiligen Schnittpunkte mit der maximalen Pumpenkennlinie $H_{n_{\text{max}}}(\dot{V})$ gegeben als $(\dot{V}_{a_{\text{max}}}, H_{a_{\text{max}}})$ und $(\dot{V}_{a_{\text{min}}}, H_{a_{\text{min}}})$. Ganz analog sei der Schnittpunkt der Anlagenkennlinie, die sich mithilfe von $a_{\text{sys}}(t_\tau)$ aufstellen lässt, gegeben als $(\dot{V}_{a_{\text{sys}}}, H_{a_{\text{sys}}})$. Wenn der Volumenstrom und die Förderhöhe nun gemäß

$$\dot{V} \rightarrow \frac{\dot{V} - \dot{V}_{a_{\text{max}}}}{\dot{V}_{a_{\text{min}}} - \dot{V}_{a_{\text{max}}}} \quad \text{und} \quad H \rightarrow \frac{H - H_{a_{\text{min}}}}{H_{a_{\text{max}}} - H_{a_{\text{min}}}}$$

durch dimensionslose Größen ersetzt werden, ist die Bogenlänge entlang der so

transformierten maximalen Pumpenkennlinie zwischen den transformierten Punkten $(\dot{V}_{a_{\max}}, H_{a_{\max}}) \rightarrow (0, 1)$ und $(\dot{V}_{a_{\text{sys}}}, H_{a_{\text{sys}}}) \rightarrow (\frac{\dot{V}_{a_{\text{sys}}} - \dot{V}_{a_{\max}}}{\dot{V}_{a_{\min}} - \dot{V}_{a_{\max}}}, \frac{H_{a_{\text{sys}}} - H_{a_{\min}}}{H_{a_{\max}} - H_{a_{\min}}})$ gerade von der Form

$$L(a_{\text{sys}}) := \int_0^{\frac{\dot{V}_{a_{\text{sys}}} - \dot{V}_{a_{\max}}}{\dot{V}_{a_{\min}} - \dot{V}_{a_{\max}}}} \sqrt{1 + \left(H'_{n_{\max}}(w \cdot (\dot{V}_{a_{\min}} - \dot{V}_{a_{\max}}) + \dot{V}_{a_{\max}}) \cdot \frac{\dot{V}_{a_{\min}} - \dot{V}_{a_{\max}}}{H_{a_{\max}} - H_{a_{\min}}} \right)^2} dw. \quad (80)$$

Mithilfe der Funktion L ist das Längenverhältnis $L(a_{\text{sys}}) / L(a_{\min})$ berechenbar, welches wiederum bei geeigneter Wahl zweier Parameter l_1 und l_0 durch

$$\tilde{\alpha}(t_\tau) := \frac{L(a_{\text{sys}}(t_\tau))}{L(a_{\min})} \cdot l_1 + l_0 \quad (81)$$

in guter Näherung mit dem unbekanntem Ventilöffnungsgrad $\alpha(t_\tau)$ in Verbindung gebracht werden kann. Für die zweite Komponente von S_τ soll schließlich

$$(S_\tau)_2 := 2 \cdot \tilde{\alpha}(t_\tau) - 1 = 2 \cdot \left(\frac{L(a_{\text{sys}}(t_\tau))}{L(a_{\min})} \cdot l_1 + l_0 \right) - 1 \quad (82)$$

gelten.

Zudem spiegeln alle möglichen Aktionen $\mathcal{A}(S_\tau)$, die ein Agent im Zustand $S_\tau \in \mathcal{S}$ für das Benchmark-System ausführen können soll, Änderungen der Drehzahl der Pumpe wider. Dabei wird festgelegt, dass der Agent sich in jedem Zeitschritt τ für eine Drehzahländerung \tilde{A}_τ aus dem Intervall $[\Delta n_{\min}^{\text{AI}}, \Delta n_{\max}^{\text{AI}}]$ mit $\Delta n_{\min}^{\text{AI}} < 0 \frac{1}{\text{min}}$ und $\Delta n_{\max}^{\text{AI}} > 0 \frac{1}{\text{min}}$ unter Berücksichtigung der minimalen und maximalen Drehzahl entscheiden kann, das heißt es gilt $\tilde{A}_\tau \in \tilde{\mathcal{A}}(S_\tau)$ mit

$$\tilde{\mathcal{A}}(S_\tau) := [\max(\Delta n_{\min}^{\text{AI}}, n_{\min} - n_{\text{pump}}(t_\tau)), \min(\Delta n_{\max}^{\text{AI}}, n_{\max} - n_{\text{pump}}(t_\tau))],$$

wobei nach Gleichung (77) die Relation

$$n_{\text{pump}}(t_\tau) = n_{\min} + \frac{1}{2} \cdot ((S_\tau)_1 + 1) \cdot (n_{\max} - n_{\min})$$

erfüllt ist. Nach Transformation auf das Intervall $[-1, 1]$ ergibt sich somit für die Menge aller möglichen Aktionen im Zustand S_τ die Definition

$$\mathcal{A}(S_\tau) := \left\{ 2 \cdot \frac{\tilde{A}_\tau - \Delta n_{\min}^{\text{AI}}}{\Delta n_{\max}^{\text{AI}} - \Delta n_{\min}^{\text{AI}}} - 1 \mid \tilde{A}_\tau \in \tilde{\mathcal{A}}(S_\tau) \right\}. \quad (83)$$

Nachdem nun sowohl die Zustände, welche die Umgebung annehmen kann, als auch die Aktionen, die der Agent durchführen kann, definiert worden sind, fehlt für die Komplementierung der Formulierung des MEPs lediglich die ma-

thematische Beschreibung der Belohnung $R_{\tau+1}$, die der Agent für die Durchführung der Aktion A_τ erhält. Belohnungen spielen eine Schlüsselrolle für das Lernverhalten des Agenten. Ein Grundprinzip, das beim Aufstellen eines effektiven Ausdrucks für die Belohnung unbedingt zu beachten ist, besagt, dass der Fokus darauf liegen muss, welches Ziel der Agent erreichen soll und nicht darauf, wie er das Ziel erreichen soll [124, S. 53f. (Abschn. 3.2)]. Für die in dieser Arbeit untersuchte Problemstellung ist hinsichtlich dessen jedoch kaum etwas falsch zu machen. Denn die zu minimierende Zielfunktion

$$\rho_{wg} \cdot \int_{t_0}^{t_f} H(t) \cdot \dot{V}(t) dt \rightarrow \min \quad (84)$$

sowie die einzuhaltende Nebenbedingung

$$\max_{t \in [t_0, t_f]} (T_{\text{set}}(t) - T_{\text{room}}(t)) - T_{\text{tol}} \leq 0, \quad (85)$$

die Teil des zu lösenden Steuerproblems (50) sind, geben ganz direkt vor, welches genaue Ziel durch die Regelung der Drehzahl der Pumpe verfolgt wird. Deswegen ist es sinnvoll die Belohnung $R_{\tau+1}$, die der Agent erhält, wenn er zur Zeit t_τ^{AI} die Entscheidung A_τ trifft, an den oben stehenden zwei Formeln zu bemessen. Dabei bietet sich zunächst eine separate Betrachtung an, aus der zwei unabhängige Werte $R_{\tau+1}^{(1)}$ und $R_{\tau+1}^{(2)}$ hervorgehen, welche anschließend in geeigneter Weise zu der Gesamtbelohnung $R_{\tau+1}$ kombiniert werden.

Für Formel (84) ist das Ziel, das Integral über die hydraulische Leistung kleinzuhalten, was tendenziell eher bei niedrigeren Drehzahlen erreicht werden kann. Für ein fiktives Szenario mit einer gleich bleibenden Anlagenkennlinie gilt nämlich, dass sich der Beitrag für das Integral mit sinkender Drehzahl ebenfalls verringert. Diese Feststellung bildet die Grundlage für die Definition

$$R_{\tau+1}^{(1)} := \frac{R_{\text{max}}}{2} \cdot (1 - (S_{\tau+1})_1), \quad (86)$$

wobei R_{max} ein Parameter ist, mit dem die Größenordnung von $R_{\tau+1}^{(1)}$ sowie weiter unten $R_{\tau+1}^{(2)}$ beeinflussbar ist.

Für die Bedingung (85) sieht die Sache ein wenig schwieriger aus, da nach Voraussetzung weder die Temperatur $T_{\text{set}}(t)$ noch $T_{\text{room}}(t)$ als bekannt gelten. Es ist demnach nicht möglich einen sinnvollen Term für $R_{\tau+1}^{(2)}$ unmittelbar auf Basis dieser zwei Größen anzugeben. Das physikalische Verständnis darüber, was sich hinter der Ungleichung (85) verbirgt, hilft jedoch einen indirekten Weg zu finden. Die Ungleichung stellt den Komfort des Nutzers sicher, indem sie dafür sorgt, dass die Raumtemperatur im Rahmen der vorgegebenen Toleranzgrenze nie signifikant unterhalb der Solltemperatur liegt. Ein notwendiges Kriterium dafür, dass diese Bedingung verletzt zu werden droht, ist ein sich

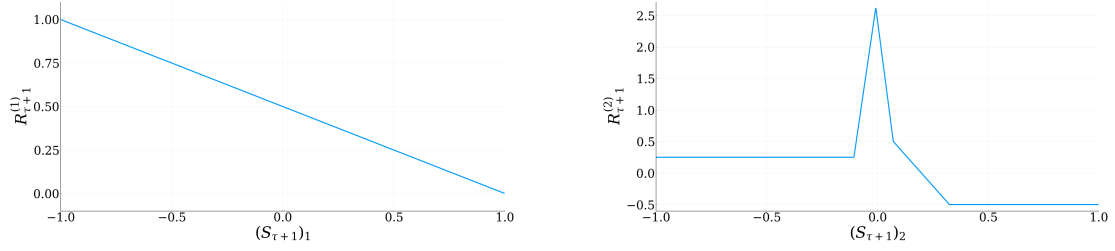


Abb. 7.1: Belohnungen $R_{\tau+1}^{(1)}$ und $R_{\tau+1}^{(2)}$ in Abhängigkeit von $(S_{\tau+1})_1$ bzw. $(S_{\tau+1})_2$

öffnendes Ventil. Das verrät ein erneuter Blick auf Abbildung 4.10, welche die typische Form einer Ventilkennlinie in Erinnerung ruft. Für alle Szenarien, in denen die Raumtemperatur die Wunschttemperatur des Nutzers unterschreitet, ist der Ventilöffnungsgrad $\alpha(t) \gg 0$ und nähert sich mit größer werdender Differenz ΔT streng monoton steigend seinem Maximalwert von 1.0, welcher für ein vollständig geöffnetes Ventil steht. Obwohl der Ventilöffnungsgrad $\alpha(t)$ nach Voraussetzung ebenfalls als unbekannt anzusehen ist, lässt sich zumindest eine Näherung über Formel (81) ermitteln. Diese kann verwendet werden, um einen passenden Term für $R_{\tau+1}^{(2)}$ aufzustellen:

$$R_{\tau+1}^{(2)} := \begin{cases} \frac{1}{4} R_{\max} & \text{für } \tilde{\alpha}(t_{\tau+1}) \leq \alpha_1, \\ \frac{19}{8} R_{\max} \cdot \frac{\tilde{\alpha}(t_{\tau+1}) - \alpha_1}{\alpha_2 - \alpha_1} + \frac{1}{4} R_{\max} & \text{für } \alpha_1 \leq \tilde{\alpha}(t_{\tau+1}) \leq \alpha_2, \\ -\frac{17}{8} R_{\max} \cdot \frac{\tilde{\alpha}(t_{\tau+1}) - \alpha_2}{\alpha_3 - \alpha_2} + \frac{21}{8} R_{\max} & \text{für } \alpha_2 \leq \tilde{\alpha}(t_{\tau+1}) \leq \alpha_3, \\ -R_{\max} \cdot \frac{\tilde{\alpha}(t_{\tau+1}) - \alpha_3}{\alpha_4 - \alpha_3} + \frac{1}{2} R_{\max} & \text{für } \alpha_3 \leq \tilde{\alpha}(t_{\tau+1}) \leq \alpha_4, \\ -\frac{1}{2} R_{\max} & \text{für } \alpha_4 \leq \tilde{\alpha}(t_{\tau+1}). \end{cases} \quad (87)$$

Dabei sind α_1 , α_2 , α_3 und α_4 durch die Ventilkennlinie vorgegebene, feste Parameter. Die Diagramme aus Figur 7.1 zeigen die Belohnungen $R_{\tau+1}^{(1)}$ und $R_{\tau+1}^{(2)}$ in Abhängigkeit der Zustandskomponenten $(S_{\tau+1})_1$ beziehungsweise $(S_{\tau+1})_2$.

Die Belohnungen $R_{\tau+1}^{(1)}$ und $R_{\tau+1}^{(2)}$ stehen sich, was die Drehzahländerungen, für die sich der Agent entscheiden muss, fast diametral gegenüber. Drehzahlabsenkungen und daraus resultierende niedrigere Drehzahlen begünstigen nämlich tendenziell recht hohe Belohnungen bezüglich $R_{\tau+1}^{(1)}$ aus Formel (86), sorgen jedoch eher für ein sich öffnendes Ventil und sind daher unvorteilhaft für die Belohnung $R_{\tau+1}^{(2)}$ aus (87). Exakt umgekehrt verhält es sich bei Drehzahlerhöhungen und den damit einhergehenden größeren Drehzahlen. Diese wirken sich negativ auf die Belohnung in (86) aus, verursachen voraussichtlich aber ein Schließen des Ventils, was einen positiven Effekt auf die Belohnung in (87) hat. Ergo steht der Agent vor der Herausforderung den richtigen Balanceakt zu vollziehen, wenn es darum geht Entscheidungen zu treffen, mit denen so viele künftige Belohnungen wie möglich eingestrichen werden können. Was

genau einen guten Balanceakt ausmacht, hängt maßgeblich davon ab, wie die beiden einzelnen Belohnungsterme $R_{\tau+1}^{(1)}$ und $R_{\tau+1}^{(2)}$ zu dem Gesamtterm $R_{\tau+1}$ zusammengeführt werden. Das ist im vorliegenden Fall etwas herausfordernd, weil es beim ersten Teilziel im Wesentlichen um das Einsparen von Energie geht, wohingegen beim zweiten Teilziel der Komfort des Nutzers maßgebend ist, und sich Energie und Komfort a priori nicht unmittelbar in natürlicher Weise gegeneinander abwägen lassen [85, 4:54–5:24 min]. Da durch die Definitionen beider Teilbelohnungen der Bezug zur Energie und Komfort jeweils bereits durch eine dimensionslose Größe derselben Größenordnung ausgedrückt wurde, besteht die verbleibende Problematik jedoch nur noch darin, die beiden Größen zu kombinieren. Üblicherweise wird diesbezüglich eine Konvexkombination der Form

$$R_{\tau+1} := (1 - \beta_R) \cdot R_{\tau+1}^{(1)} + \beta_R \cdot R_{\tau+1}^{(2)} \quad (88)$$

mit einem Gewichtungsfaktor $\beta_R \in [0, 1]$ verwendet [89, S. 349 (Gl. (1))].

7.2 Testphase

Ein Merkmal der neuen Strategie ist, dass während der Steuerphase mithilfe einer Näherung an die optimale Action-Value-Funktion stets eine Bewertung des aktuellen Systemzustandes mit Blick auf Unter- oder Überversorgungen vorgenommen wird, auf deren Grundlage der Agent gemäß Gleichung (76) seine Entscheidungen für Drehzahländerungen trifft. Um eine Beurteilung des Systemzustandes hinsichtlich der Versorgungsgüte während der Steuerphase vornehmen zu können, ist eine vorab durchgeführte Testphase hilfreich, in der kurzzeitig ganz bewusst diverse Drehzahlen angefahren und so unterschiedliche Systemzustände erzeugt werden. Die im Folgenden beschriebene Testphase, in die der Nutzer durch sein Verhalten idealerweise gar nicht eingreifen sollte, dient im Wesentlichen zwei Zwecken. Zum einen sollen Näherungen für den maximalen und den minimalen Parabelparameter a_{\max} und a_{\min} aus den Anlagenkennlinien (78) und (79) ermittelt werden, sodass über das Kennfeld der Pumpe auch Näherungen für $\dot{V}_{a_{\min}}$, $\dot{V}_{a_{\max}}$, $H_{a_{\max}}$ und $H_{a_{\min}}$ berechenbar werden, wodurch wiederum die Funktion $L(a_{\text{sys}})$ aus Formel (80) näherungsweise ebenfalls zur Verfügung steht. Zum anderen sollen Datenpunkte gesammelt werden, mit denen am Ende der Testphase, also unmittelbar vor der eigentlichen Steuerphase, ein neuronales Netz für die Näherung an die optimale Action-Value-Funktion trainiert wird. Aus diesem Grund setzt sich die Testphase aus zwei Teilen zusammen. Im ersten Teil wird für eine Gesamtdauer von $\Delta t_{\text{mxmn}}^{\text{AI}}$ sowohl die minimale als auch die maximale Drehzahl der Pumpe gezielt angefahren. In diesem auch als n_{\max} - n_{\min} -Phase bezeichnetem Abschnitt, wird

für die Steuergröße

$$u^{\text{AI}}(t) := \begin{cases} n_{\text{pump},0} + \frac{t-t_0}{\Delta t_{\text{trans}}^{\text{AI}}} \cdot (n_{\text{max}} - n_{\text{pump},0}) & \text{für } t_0 \leq t \text{ und} \\ & t \leq t_0 + \Delta t_{\text{trans}}^{\text{AI}}, \\ n_{\text{max}} & \text{für } t_0 + \Delta t_{\text{trans}}^{\text{AI}} \leq t \\ & \text{und } t \leq t_0 + 0.7\Delta t_{\text{mxmn}}^{\text{AI}}, \\ n_{\text{max}} - \frac{t-(t_0+0.7\Delta t_{\text{mxmn}}^{\text{AI}})}{\Delta t_{\text{trans}}^{\text{AI}}} \cdot (n_{\text{max}} - n_{\text{min}}) & \text{für } t_0 + 0.7\Delta t_{\text{mxmn}}^{\text{AI}} \leq t \text{ und} \\ & t \leq t_0 + 0.7\Delta t_{\text{mxmn}}^{\text{AI}} + \Delta t_{\text{trans}}^{\text{AI}}, \\ n_{\text{min}} & \text{für } t_0 + 0.7\Delta t_{\text{mxmn}}^{\text{AI}} + \Delta t_{\text{trans}}^{\text{AI}} \leq t \\ & \text{und } t \leq t_0 + \Delta t_{\text{mxmn}}^{\text{AI}}, \end{cases}$$

festgelegt, wobei $\Delta t_{\text{trans}}^{\text{AI}}$ eine kurze Zeitspanne angibt, in der stetige Übergänge von einer Drehzahl zur nächsten stattfinden. Die gesuchten Näherungen für den maximalen und minimalen Parabelparameter ergeben sich bei der Auswertung während der $n_{\text{max}}-n_{\text{min}}$ -Phase durch

$$\begin{aligned} \tilde{a}_{\text{max}} &:= \max_{t \in [t_0, t_0 + 0.7\Delta t_{\text{mxmn}}^{\text{AI}}]} H(t) / \dot{V}(t)^2 \text{ und} \\ \tilde{a}_{\text{min}} &:= \min_{t \in [t_0 + 0.7\Delta t_{\text{mxmn}}^{\text{AI}}, t_0 + \Delta t_{\text{mxmn}}^{\text{AI}}]} H(t) / \dot{V}(t)^2. \end{aligned} \quad (89)$$

Der zweite Teil der Testphase, die Haupttestphase, schließt unmittelbar an die $n_{\text{max}}-n_{\text{min}}$ -Phase an. Beginnend bei der minimalen Drehzahl werden immer größer werdend und mit äquidistanten Abstand $K_{\text{tp}}^{\text{AI}}$ weitere Bezugsdrehzahlen angefahren. Diese $K_{\text{tp}}^{\text{AI}} + 1$ Drehzahlen wiederum dienen als Ausgangspunkt für jeweils $J_{\text{tp}}^{\text{AI}}$ Tests der Dauer $\Delta t_{\text{tp}}^{\text{AI}}$, welche übergangslos nacheinander durchgeführt werden, bevor es zu einem $\Delta t_{\text{trans}}^{\text{AI}}$ andauernden Übergang zur nächsten Bezugsdrehzahl kommt. Die Definitionen

$$\begin{aligned} t_{\text{test}}^{(k,0)} &:= t_0 + \Delta t_{\text{mxmn}}^{\text{AI}} + k \cdot J_{\text{tp}}^{\text{AI}} \cdot \Delta t_{\text{tp}}^{\text{AI}} + k \cdot \Delta t_{\text{trans}}^{\text{AI}} \\ t_{\text{test}}^{(k,j)} &:= t_0 + \Delta t_{\text{mxmn}}^{\text{AI}} + (k \cdot J_{\text{tp}}^{\text{AI}} + j) \cdot \Delta t_{\text{tp}}^{\text{AI}} + k \cdot \Delta t_{\text{trans}}^{\text{AI}} \end{aligned}$$

für $k \in \{0, 1, \dots, K_{\text{tp}}^{\text{AI}}\}$ und $j \in \{1, 2, \dots, J_{\text{tp}}^{\text{AI}}\}$ markieren die Zeitpunkte, an denen eine Bezugsdrehzahl

$$n_{\text{test}}^{(k,0)} := n_{\text{min}} + k \cdot \frac{n_{\text{max}} - n_{\text{min}}}{K_{\text{tp}}^{\text{AI}}}$$

angenommen wird, beziehungsweise ein Test mit der zugehörigen Drehzahl

$$n_{\text{test}}^{(k,j)} := n_{\text{min}} + k \cdot \frac{n_{\text{max}} - n_{\text{min}}}{K_{\text{tp}}^{\text{AI}}} + \sum_{i=1}^j (C_{\text{tp}}^{\text{AI}})^{(k+1,i)} \cdot \frac{\Delta n_{\text{max}}^{\text{AI}} - \Delta n_{\text{min}}^{\text{AI}}}{2 \cdot (K_{\text{tp}}^{\text{AI}} + 1)}$$

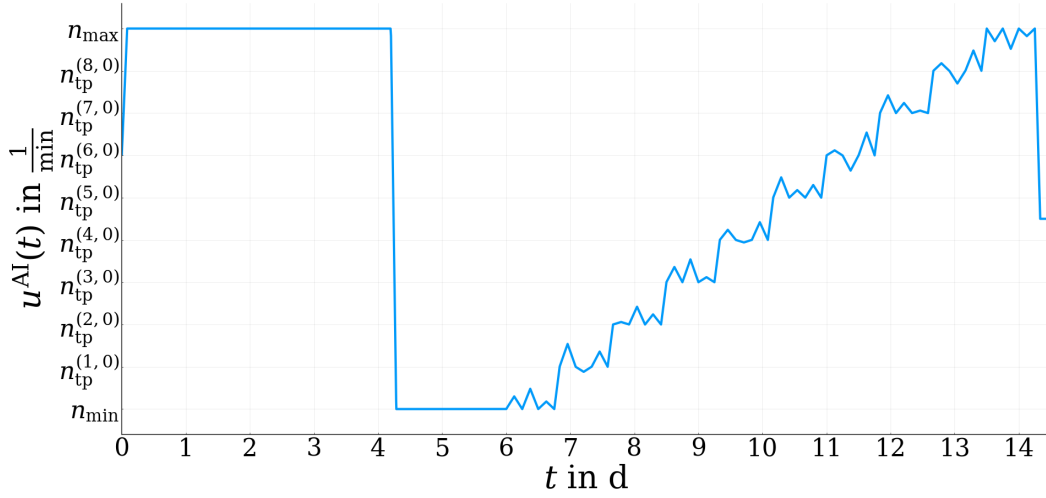


Abb. 7.2: Drehzahlsteuerung der neuen Strategie während der Testphase

endet. Dabei ist $(C_{\text{tp}}^{\text{AI}})^{(k,j)}$ der Eintrag einer Matrix mit ganzzahligen Koeffizienten aus der Menge $\{-K_{\text{tp}}^{\text{AI}}, \dots, K_{\text{tp}}^{\text{AI}}\}$, die so zu parametrieren ist, dass die angefahrenen Drehzahlen möglichst unterschiedliche Szenarien abbilden. Das ist etwa dadurch zu erreichen, dass in jeder Zeile sowohl negative als auch positive Zahlen sowie sowohl betragsmäßig große als auch betragsmäßig kleine stehen. Die Matrix, die hier verwendet wird und diese Bedingungen erfüllt, befindet sich im Anhang C.10. Die Steuergröße nimmt in der Haupttestphase für $k \in \{0, 1, \dots, K_{\text{tp}}^{\text{AI}}\}$ schließlich die Gestalt

$$u^{\text{AI}}(t) := \begin{cases} n_{\text{test}}^{(k,j-1)} + \frac{n_{\text{test}}^{(k,j)} - n_{\text{test}}^{(k,j-1)}}{\Delta t_{\text{tp}}^{\text{AI}}} \cdot (t - t_{\text{test}}^{(k,j-1)}) & \text{für } t_{\text{test}}^{(k,j-1)} \leq t \leq t_{\text{test}}^{(k,j)} \text{ für alle } j \in \{1, \dots, J_{\text{tp}}^{\text{AI}}\}, \\ n_{\text{test}}^{(k, J_{\text{tp}}^{\text{AI}})} + \frac{n_{\text{test}}^{(k+1,0)} - n_{\text{test}}^{(k, J_{\text{tp}}^{\text{AI}})}}{\Delta t_{\text{trans}}^{\text{AI}}} \cdot (t - t_{\text{test}}^{(k, J_{\text{tp}}^{\text{AI}})}) & \text{für } t_{\text{test}}^{(k, J_{\text{tp}}^{\text{AI}})} \leq t \leq t_{\text{test}}^{(k, J_{\text{tp}}^{\text{AI}})} + \Delta t_{\text{trans}}^{\text{AI}}, \\ n_{\text{test}}^{(K_{\text{tp}}^{\text{AI}}+1,0)} & \text{für } t_{\text{test}}^{(K_{\text{tp}}^{\text{AI}}, J_{\text{tp}}^{\text{AI}})} + \Delta t_{\text{trans}}^{\text{AI}} \leq t \leq t_{\text{test}}^{(K_{\text{tp}}^{\text{AI}}, J_{\text{tp}}^{\text{AI}})} + 3\Delta t_{\text{trans}}^{\text{AI}}, \end{cases}$$

an und wird somit dadurch beendet, dass die durch

$$n_{\text{test}}^{(K_{\text{tp}}^{\text{AI}}+1,0)} := \frac{1}{2} \cdot (n_{\text{min}} + n_{\text{max}})$$

definierte Drehzahl angesteuert wird. Die Abbildung 7.2 zeigt sowohl für die $n_{\text{max}} - n_{\text{min}}$ -Phase als auch exemplarisch für die Haupttestphase mit $K_{\text{tp}}^{\text{AI}} = 9$ und $J_{\text{tp}}^{\text{AI}} = 6$ den Verlauf der Steuergröße.

7.3 Initialisierung der Steuerphase

Die Durchführung der im vorherigen Abschnitt beschriebenen Haupttestphase dient vor allem dem Sammeln von Daten, die für die Initialisierung der eigentlichen Steuerphase von Bedeutung sind. Bevor gezeigt wird, welche Daten genau gesammelt und welche Hilfsgrößen damit berechnet werden, ist es sinnvoll zunächst das Kernelement der eigentlichen Steuerphase einzuführen. Bei diesem handelt es sich um eine Näherung $\tilde{q}_*(s, a)$ der optimalen Action-Value-Funktion aus Formel (75), welche durch ein flaches neuronales Netz aus L Schichten parametrisiert wird, wobei die mathematische Vorgehensweise in diesem Abschnitt sich an die Darstellung von Ng [104], [106], [105] anlehnt. Die Eingangsschicht des neuronalen Netzes, die nicht mit zu den insgesamt L Schichten zählt, sondern als 0. Schicht anzusehen ist, besteht aus $n^{[0]} := 3$ Neuronen und die Ausgangsschicht aus $n^{[L]} := 1$ Neuron. Für die weitere Architektur des Netzes erweist es sich als ausreichend, $L = 3$ zu wählen, also nur zwei verborgene Schichten zu nutzen. Das Potenzial eines dreischichtigen neuronalen Netzes gilt nämlich für viele praktische Probleme bereits als sehr hoch [51, S. 157f. u. S. 158 (Tab. 5.1)]. Hinsichtlich der Wahl von $n^{[1]}$ und $n^{[2]}$ existieren eine Reihe von Empfehlungen, jedoch laufen mit $n^{[0]} = 3$ und $n^{[L]} = 1$ fast alle Orientierungsregeln wie $n^{[1]} + n^{[2]} = n^{[0]} + n^{[L]} - \frac{1}{2}$ oder $n^{[1]} + n^{[2]} < 2 \cdot n^{[0]}$ auf zu kleine Netzstrukturen wie etwa $n^{[1]} = 2$ und $n^{[2]} = 3$ hinaus [121, S. 2 ff. (Abschn. 2)]. Weil demgegenüber in der Theorie neuronale Netze mit $n^{[1]} = n^{[0]} = 3$ und $n^{[2]} = 2 \cdot n^{[0]} + 2 = 8$ konstruiert werden könnten, die beliebig gute Approximationen darstellen [60, S. 963 (Abstract)], wird hier als eine Art Kompromiss $n^{[1]} := 4$ und $n^{[2]} := 6$ gesetzt. Ferner seien die Aktivierungen der Neuronen der verborgenen Schichten mit $h_j^{[l]}$ für $j \in \{1, \dots, n^{[l]}\}$ und $l \in \{1, \dots, L-1\}$ bezeichnet, das Gewicht der Verbindung des i -ten Neurons der $(l-1)$ -ten Schicht mit dem j -ten Neuron der l -ten Schicht trage den Namen $W_{ji}^{[l]}$ und der additive Bias heiße $b_j^{[l]}$ für $l \in \{1, \dots, L\}$. Dabei sind $h^{[l]} \in \mathbb{R}^{n^{[l]}}$ sowie $b^{[l]} \in \mathbb{R}^{n^{[l]}}$ auch als Vektoren und $W^{[l]} \in \mathbb{R}^{n^{[l]} \times n^{[l-1]}}$ auch als Matrizen aufzufassen. Das neuronale Netz, das schematisch in der Skizze 7.3 zu sehen ist, beschreibt nach Belegung seiner Parameter $W^{[l]}$, $b^{[l]}$ für $l \in \{1, \dots, L\}$ eine im allgemeinen Fall nichtlineare Abbildung, welche den Eingangsvektor e dem Ausgangsskalar \tilde{q}_* zuordnet. Die dem neuronalen Netz zugrunde liegende Berechnungsvorschrift lautet dabei in Vektorschreibweise

$$\left. \begin{aligned} z^{[1]} &:= W^{[1]}e + b^{[1]} \\ h^{[l]} &:= \sigma^{[l]}(z^{[l]}) \\ z^{[l+1]} &:= W^{[l+1]}h^{[l]} + b^{[l+1]} \end{aligned} \right\} l \in \{1, \dots, L-1\} \quad (90)$$

$$\tilde{q}_* := \sigma^{[L]}(z^{[L]})$$

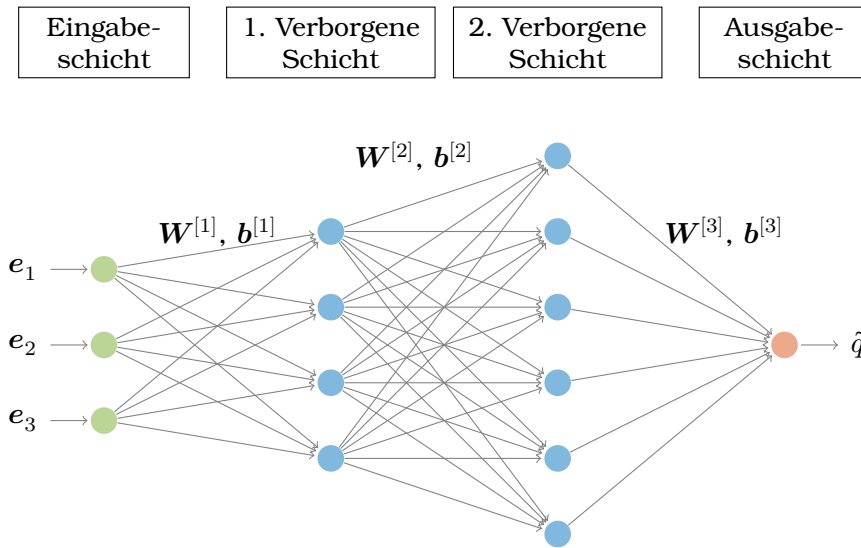


Abb. 7.3: Skizze des neuronalen Netzes für die Näherung der optimalen Action-Value-Funktion

[43, S. 205 (Algorithm 6.3)]. Für die vektorwertigen bzw. ein wenig weiter unten matrixwertigen Aktivierungsfunktionen $\sigma^{[l]}$ mit $l \in \{1, \dots, L\}$ wird dem modernen Ansatz folgend der Rectifier

$$\sigma(v) = \max(0, v)$$

benutzt [2, S. 13], der bei vektorwertigen Argumenten $v \in \mathbb{R}^{n^{[l]}}$ gemäß $\sigma_j^{[l]}(v) := \sigma(v_j)$ bzw. bei matrixwertigen Argumenten $M \in \mathbb{R}^{n^{[l]} \times n^{[l-1]}}$ gemäß $\sigma_{ji}^{[l]}(M) := \sigma(M_{ji})$ für $l \in \{1, \dots, L\}$, $j \in \{1, \dots, n^{[l]}\}$ und $i \in \{1, \dots, n^{[l-1]}\}$ jeweils komponentenweise wirkt. Der Funktionswert der parametrisierten Näherung von $\tilde{q}_*(s, a)$ sei nun gerade dadurch gegeben, dass für $e = (s, a)$ der Output des neuronalen Netzes anhand der Formeln (90) ermittelt wird.

Die Güte der Näherung beeinflusst die Performance der neuen Steuerstrategie maßgeblich. Deshalb geht der eigentlichen Steuerphase auch die im vorherigen Abschnitt beschriebene Testphase voraus, in der Daten gesammelt werden, mit denen es möglich ist, bereits vorab eine günstige Belegung der Parameter zu ermitteln. Die etwas weiter unten präsentierte Verarbeitung der Ergebnisse der Testphase führt zu einer Verbesserung der Approximation $\tilde{q}_*(s, a)$, indem das in der Testphase erlernte Verhalten des Systems durch Anpassung der Parameter trainiert wird. Eine gängige Wahl für die anfängliche Belegung der Parameter eines neuronalen Netzes, die sich in der Praxis vielfach bewährt

hat, stellt die He-Initialisierung

$$\begin{aligned}
 \mathbf{b}^{[l]} &= \mathbf{0} \quad \text{für } l \in \{1, \dots, L\}, \\
 \mathbf{W}_{ji}^{[l]} &\sim \mathcal{N}(0, 1) \cdot \sqrt{\frac{2}{n^{[l-1]} \cdot n^{[l]}}} \quad \text{für } l \in \{1, \dots, L\}, \\
 & \quad j \in \{1, \dots, n^{[l]}\}, i \in \{1, \dots, n^{[l-1]}\}
 \end{aligned} \tag{91}$$

dar, bei der mithilfe der Standardnormalverteilung $\mathcal{N}(0, 1)$ zufällige Werte ausgewählt werden [72, S. 212 (Kap. 9)]. Ein solches Vorgehen ließe jedoch jegliches vorhandenes Wissen über das System außer Acht und ist deswegen eher unvorteilhaft. Vielmehr bietet es sich an, eine Initialisierung vorzunehmen, die bereits mit einer akzeptablen Näherung für die Action-Value-Funktion korrespondiert. Eine derartige Vorinitialisierung lässt sich dadurch konstruieren, dass eine einfache Steuerstrategie erdacht wird, eine dazu passende, fiktive Belohnungsfunktion aufgestellt wird und mit der wiederum die zugehörige, fiktive, optimale Action-Value Funktion berechnet wird. Diese kann dann dazu benutzt werden, Wertepaare aufzustellen, mit denen das neuronale Netz trainiert und so die Vorinitialisierung bestimmt wird. Als fiktive Steuerstrategie bietet es sich an, sich für diejenige Strategie zu entscheiden, bei der die Pumpe möglichst durchgehend mit der konstanten mittleren Drehzahl $\frac{1}{2} \cdot (n_{\min} + n_{\max})$ betrieben wird. Eine Belohnungsfunktion, die diesem Ziel entspreche und nur von der ersten Komponente der durch Gleichung (77) und (82) beschriebenen Systemzustände abhinge, wäre etwa $\hat{R}_{\tau+1} := R_{\max} \cdot (1 - |(S_{\tau+1})_1|)$. Bei der optimalen Policy $\hat{\pi}_*$, welche sich auf diesen Gedankengang bezieht, würde der Agent demnach in jedem Zustand die Drehzahl stets dahingehend verändern, dass er so dicht wie möglich an die angestrebte mittlere Drehzahl kommt bzw. wenn er diese erreicht hat, keine weiteren Änderungen mehr vornehmen. Die Durchschnittsbelohnung dieser optimalen Policy wäre laut Definition (70) dann gerade $\hat{R}(\hat{\pi}_*) = R_{\max}$. Weil $\hat{\pi}_*$ deterministisch ist, könnte der Wert der zugehörigen optimalen Action-Value-Funktion $\hat{q}_*(s, a)$ auf Basis von Formel (73) als eine endliche Summe von Differenzen berechnet werden. Der Anhang D.5 zeigt den Quellcode einer Julia-Funktion, deren Rückgabewert dem Funktionswert von \hat{q}_* entspricht.

Für diese fiktive Funktion soll nun ein neuronales Netz trainiert werden mit dem Ziel, die trainierten Parameter als Vorinitialisierung für das neuronale Netz von $\tilde{q}_*(s, a)$ übernehmen zu können. Weil in dem erfundenen Szenario die Action-Value-Funktion jedoch effektiv nur von der ersten Komponente des Systemzustandes aus (77) abhängt, wird zunächst nur mit $\tilde{n}^{[0]} = 2$ gerechnet. Das weggenommene Neuron, welches die zweite Komponente des Zustandes aus (82) repräsentiert, kann nämlich für die gesuchte Vorinitialisierung leicht wieder hinzugefügt werden, indem eine Nullspalte in das Ergebnis für $\mathbf{W}^{[1]}$ eingefügt wird. Zur Generierung eines Trainingsdatensatzes sei durch jeweilige

äquidistante Unterteilung des Intervalls $[-1, 1]$ eine Menge $\{s_i \mid i \in \{1, \dots, N_s\}\}$ von N_s Testzuständen und eine Menge $\{a_i \mid i \in \{1, \dots, N_a\}\}$ von N_a möglichen Testaktionen gegeben. Weiter sei N die Anzahl aller Testpaare der Form

$$e^{(i)} := (s_i, a_i) \quad \text{für } i \in \{1, \dots, N\}, \quad (92)$$

die sich daraus unter Berücksichtigung von $a_i \in \mathcal{A}(s_i)$ kombinieren lässt. Die Zuweisungen

$$q^{(i)} := \hat{q}_*(s_i, a_i) \quad \text{für } i \in \{1, \dots, N\} \quad (93)$$

vervollständigen schließlich einen Testdatensatz und finden sich ihrerseits in der Definition der mittleren quadratischen Fehlerverlustfunktion

$$\begin{aligned} \mathcal{C}(\mathbf{b}^{[1]}, \mathbf{W}^{[1]}, \dots, \mathbf{b}^{[L]}, \mathbf{W}^{[L]}) &:= \frac{1}{N} \cdot \sum_{i=1}^N \mathcal{L}(\tilde{q}_*^{(i)}, q^{(i)}) \\ \text{mit } \mathcal{L}(\tilde{q}_*, q) &:= |\tilde{q}_* - q|^2 \end{aligned} \quad (94)$$

wieder [43, S. 105f. (Formel (5.4))]. Jene beschreibt den Mittelwert der quadratischen Abweichungen zwischen den berechneten Ausgabeskalaren $\tilde{q}_*^{(i)}$ des neuronalen Netzes nach Vorschrift (90) bei Verwendung der entsprechenden Eingangsvektoren $e^{(i)}$ aus Gleichung (92), und den tatsächlichen, gewünschten Werten $q^{(i)}$ des generierten Trainingsdatensatzes aus Formel (93). Demzufolge liefern die Argumente, welche die Fehlerverlustfunktion minimieren, gerade die für die Vorinitialisierung gesuchten Parameterwerte. Da der Umfang N des generierten Trainingsdatensatzes hinreichend klein auswählbar ist, besteht hinsichtlich des Optimierungsprozesses kein spezieller Anlass dafür, nicht in jedem Iterationsschritt vor einem Update der zu optimierenden Parameter den gesamten Batch aller generierten Trainingsdaten auf einmal zu verarbeiten [72, S. 195f. (Kap. 9)]. Diesem Zweck dienlich ist die Einführung der Matrizen $\mathbf{E} \in \mathbb{R}^{\tilde{n}^{[0]} \times N}$ und $\mathbf{Q} \in \mathbb{R}^{\tilde{n}^{[L]} \times N}$ durch

$$\mathbf{E} := (e^{(1)}, e^{(2)}, \dots, e^{(N)}) \quad \text{und} \quad \mathbf{Q} := (q^{(1)}, q^{(2)}, \dots, q^{(N)}).$$

Denn mit ihnen existiert die Möglichkeit den Output des neuronalen Netzes nach Formel (90) für den gesamten Batch durch die Forwardpropagation

$$\left. \begin{aligned} \mathbf{Z}^{[1]} &:= \mathbf{W}^{[1]} \mathbf{E} \oplus \mathbf{b}^{[1]} \\ \mathbf{H}^{[l]} &:= \sigma^{[l]}(\mathbf{Z}^{[l]}) \\ \mathbf{Z}^{[l+1]} &:= \mathbf{W}^{[l+1]} \mathbf{H}^{[l]} \oplus \mathbf{b}^{[l+1]} \\ \tilde{\mathbf{Q}}_* &:= \sigma^{[L]}(\mathbf{Z}^{[L]}) \end{aligned} \right\} \quad l \in \{1, \dots, L-1\} \quad (95)$$

zeitgleich zu berechnen und mit den Ergebnissen wiederum den Gradienten

von der Fehlerverlustfunktion (94) durch die Backpropagation

$$\left. \begin{aligned}
 \frac{\partial \mathcal{L}}{\partial \mathbf{Z}^{[L]}} &:= 2 \cdot (\tilde{\mathbf{Q}}_* - \mathbf{Q}) \cdot \left(\boldsymbol{\sigma}^{[L]'}(\mathbf{Z}^{[L]}) \right) \\
 \frac{\partial \mathcal{C}}{\partial \mathbf{W}^{[l]}} &:= \frac{1}{N} \cdot \left(\frac{\partial \mathcal{L}}{\partial \mathbf{Z}^{[l]}} \right) (\mathbf{H}^{[l-1]})^T \\
 \frac{\partial \mathcal{C}}{\partial \mathbf{b}_j^{[l]}} &:= \frac{1}{N} \cdot \sum_{i=1}^N \left(\frac{\partial \mathcal{L}}{\partial \mathbf{Z}^{[l]}} \right)_{ji} \quad \text{für } j \in \{1, \dots, n^{[l]}\} \\
 \frac{\partial \mathcal{C}}{\partial \mathbf{Z}^{[l-1]}} &:= \left((\mathbf{W}^{[l]})^T \left(\frac{\partial \mathcal{L}}{\partial \mathbf{Z}^{[l]}} \right) \right) \odot \left(\boldsymbol{\sigma}^{[l-1]'}(\mathbf{Z}^{[l-1]}) \right) \\
 \frac{\partial \mathcal{C}}{\partial \mathbf{W}^{[1]}} &:= \frac{1}{N} \cdot \left(\frac{\partial \mathcal{L}}{\partial \mathbf{Z}^{[1]}} \right) \mathbf{E}^T \\
 \frac{\partial \mathcal{C}}{\partial \mathbf{b}_j^{[1]}} &:= \frac{1}{N} \cdot \sum_{i=1}^N \left(\frac{\partial \mathcal{L}}{\partial \mathbf{Z}^{[1]}} \right)_{ji} \quad \text{für } j \in \{1, \dots, n^{[1]}\}
 \end{aligned} \right\} \quad l \in \{L, \dots, 2\} \quad (96)$$

zu bestimmen [107, S. 43–50 (Abschn. 2.3–2.6)]. Hierbei bezeichnen die Operationen $M \oplus v$ die spaltenweise Addition eines Vektors v zu jeder Spalte einer Matrix M sowie $M_1 \odot M_2$ die komponentenweise Multiplikation zweier Matrizen M_1 und M_2 .

Mit dem generierten Testdatensatz aus (92) und (93), den Anfangswerten aus (91) sowie der Forward- und Backpropagation aus (95) und (96) stehen nun alle benötigten Mittel zur Verfügung, um die Fehlerverlustfunktion aus (94) zu minimieren und somit die Vorinitialisierung des neuronalen Netzes für die Auswertung der in der Haupttestphase gesammelten Daten zu bestimmen. Weil die Resultate der Minimierung von \mathcal{C} aufgrund der Projektion der Drehzahl auf das Intervall $[-1, 1]$ unabhängig von der Wahl des Pumpenmodells sind, müssen diese Berechnungen nicht während des Pumpenbetriebes durchgeführt werden, sondern können bereits vorab mithilfe eines PCs vorgenommen werden. In dieser Arbeit geschah dies mit Julia, wobei die Wahl des Solvers zur Lösung des Minimierungsproblems auf das Adamverfahren [68] fiel. Im Anhang C.10 stehen die draus hervorgehenden Ergebnisse für die Vorinitialisierung. Dabei wurde in der zweiten Spalte von $\mathbf{W}^{[1]}$ wie weiter oben beschrieben, bereits eine Nullspalte zur weiteren Nutzung hinzugefügt.

Mit dieser Vorinitialisierung steht eine erste Approximation $\tilde{q}_*(s, a)$ für die optimale Action-Value-Funktion aus Formel (75) zur Verfügung, von der angenommen werden kann, dass sie der Pumpe bekannt ist. In dieser Approximation findet die Abhängigkeit von der zweiten Komponente des Zustandes aus Formel (82) nach Konstruktion allerdings noch keine Berücksichtigung, was der Grund für die Durchführung der im vorherigen Abschnitt beschriebenen Haupttestphase ist. Bei den zu den Zeitpunkten $t_{\text{test}}^{(k,j)}$ durchgeführten Tests können auf Basis der Förderhöhe $H(t_{\text{test}}^{(k,j)})$ und des Volumenstroms $\dot{V}(t_{\text{test}}^{(k,j)})$ nämlich die nachfolgenden Daten berechnet und gesammelt werden, mit denen

wiederum am Ende der Testphase die vorinitialisierten Parameter des neuronalen Netzes weitertrainiert werden können, um so eine günstige Initialisierung für die eigentliche Steuerphase zu ermitteln. Bei den gesammelten Daten handelt es sich zum einen um die Testzustände

$$s_{\text{tp}}^{(k,j)} := \left(2 \cdot \frac{n_{\text{test}}^{(k,j)} - n_{\text{min}}}{n_{\text{max}} - n_{\text{min}}} - 1, \quad 2 \cdot \left(\frac{L(a_{\text{sys}}(t_{\text{test}}^{(k,j)}))}{L(\tilde{a}_{\text{min}})} \cdot l_1 + l_0 \right) - 1 \right)$$

sowie die Testaktionen

$$a_{\text{tp}}^{(k,j)} := 2 \cdot \frac{(n_{\text{test}}^{(k,j+1)} - n_{\text{test}}^{(k,j)}) - \Delta n_{\text{min}}^{\text{AI}}}{\Delta n_{\text{max}}^{\text{AI}} - \Delta n_{\text{min}}^{\text{AI}}} - 1$$

für $k \in \{0, 1, \dots, K_{\text{tp}}^{\text{AI}}\}$ und $j \in \{1, 2, \dots, J_{\text{tp}}^{\text{AI}}\}$, welche aus den Gleichungen (77), (82) und (83) dadurch hervorgehen, dass die Testzeitpunkte $t_{\text{test}}^{(k,j)}$ anstelle der Steuerzeitpunkte t_{τ} treten. Die unbekanntenen Größen a_{max} und a_{min} , die auch indirekt in der Funktion $L(\cdot)$ aus (80) auftreten, werden dabei außerdem mit den in der $n_{\text{max}}-n_{\text{min}}$ -Phase ermittelten Näherungen \tilde{a}_{max} und \tilde{a}_{min} aus (89) substituiert. Ebenso wie die Testzustände und Testaktionen seien weiter die Testbelohnungen $r_{\text{tp}}^{(k,j)}$ dadurch gegeben, dass in den Formeln (86), (87) und (88) der Steuerzeitpunkt $t_{\tau+1}$ durch den entsprechenden Testzeitpunkt $t_{\text{test}}^{(k,j)}$ ersetzt wird. Weiter werden für $k \in \{0, 1, \dots, K_{\text{tp}}^{\text{AI}}\}$ und $j \in \{1, 2, \dots, J_{\text{tp}}^{\text{AI}} - 1\}$ die Hilfsgrößen

$$\delta_{\text{tp}}^{(k,j)} := r_{\text{tp}}^{(k,j+1)} - \bar{r}_{\text{tp}}^{(k,j)} + \max_{a \in \mathcal{A}(s_{\text{tp}}^{(k,j+1)})} \tilde{q}_*(s_{\text{tp}}^{(k,j+1)}, a) - \tilde{q}_*(s_{\text{tp}}^{(k,j)}, a_{\text{tp}}^{(k,j)}) \quad (97)$$

eingeführt, in denen die Durchschnittsbelohnen $\bar{r}_{\text{tp}}^{(k,j)}$ auftauchen, welche ihrerseits rekursiv durch $\bar{r}_{\text{tp}}^{(0,1)} := \frac{R_{\text{max}}}{2}$ sowie

$$\begin{aligned} \bar{r}_{\text{tp}}^{(k,j+1)} &:= \bar{r}_{\text{tp}}^{(k,j)} + \beta_{\text{tp}}^{\text{AI}} \cdot \delta_{\text{tp}}^{(k,j)} \quad \text{für } k \in \{0, 1, \dots, K_{\text{tp}}^{\text{AI}}\}, j \in \{1, 2, \dots, J_{\text{tp}}^{\text{AI}} - 2\} \\ \text{und } \bar{r}_{\text{tp}}^{(k+1,1)} &:= \bar{r}_{\text{tp}}^{(k, J_{\text{tp}}^{\text{AI}} - 1)} + \beta_{\text{tp}}^{\text{AI}} \cdot \delta_{\text{tp}}^{(k, J_{\text{tp}}^{\text{AI}} - 1)} \quad \text{für } k \in \{0, 1, \dots, K_{\text{tp}}^{\text{AI}} - 1\} \end{aligned}$$

definiert werden [133, S. 10655 (Gl. 3 u. Gl. 4)]. Dabei ist die Lernrate $\beta_{\text{tp}}^{\text{AI}} > 0$ ein Hyperparameter der Testphase. Die Hilfsgrößen aus (97) und eine weitere Lernrate $\alpha_{\text{tp}}^{\text{AI}} > 0$ als Hyperparameter ermöglichen es für $\tilde{q}_*(s_{\text{tp}}^{(k,j)}, a_{\text{tp}}^{(k,j)})$ eine verbesserte Näherung der Form

$$q_{\text{tp}}^{(k,j)} := \tilde{q}_*(s_{\text{tp}}^{(k,j)}, a_{\text{tp}}^{(k,j)}) + \alpha_{\text{tp}}^{\text{AI}} \cdot \delta_{\text{tp}}^{(k,j)}$$

zu ermitteln [133, S. 10655 (Gl. 2)]. Aus diesem Grund lassen sich die während der Haupttestphase gesammelten Daten $e_{\text{tp}}^{(k,j)} := (s_{\text{tp}}^{(k,j)}, a_{\text{tp}}^{(k,j)})$ und $q_{\text{tp}}^{(k,j)}$ für $k \in \{0, 1, \dots, K_{\text{tp}}^{\text{AI}}\}$ und $j \in \{1, 2, \dots, J_{\text{tp}}^{\text{AI}} - 1\}$ dafür nutzen, das vorinitialisier-

te neuronale Netz weiterzutrainieren und dadurch die Approximation für \tilde{q}_* zu verbessern. Das Vorgehen hierbei ähnelt dem Vorgehen zur Bestimmung der Vorinitialisierung, unterscheidet sich jedoch in zwei wesentlichen Punkten. Zum einen müssen die Berechnungen während des Betriebs der Pumpe durchgeführt werden und zum anderen sind die Werte $q_{\text{tp}}^{(k,j)}$ im allgemeinen Fall selbst nur grobe Näherungen, die gegebenenfalls noch recht weit von den gewünschten, aber unbekanntenen Werten $q_*(s_{\text{tp}}^{(k,j)}, a_{\text{tp}}^{(k,j)})$ entfernt sind. Letztere Feststellung impliziert, dass das neuronale Netz nicht zu stark auf den in der Haupttestphase bestimmten Datensatz weitertrainiert werden darf, da andernfalls ein Overfitting auf unter Umständen ungünstige Werte droht [1, S. 485 (Abschn. 1) u. S. 486 ff. (Abschn. 4)]. Um dem entgegenzuwirken wird hier eine Technik benutzt, die Early Stopping genannt wird und die Anzahl der Epochen für das Weitertraining des neuronalen Netzes mit dem Adamverfahren und dem aus der Haupttestphase stammenden Datensatz auf $N_{\text{tp}}^{\text{AI}}$ begrenzt [43, S. 239 f. (Abschn. 7.8)]. Der Hyperparameter $N_{\text{tp}}^{\text{AI}}$ darf allerdings auch nicht zu klein gewählt werden, da sonst die Abhängigkeit der zweiten Komponenten des Zustandes aus Formel (82) möglicherweise nicht ausreichend in der Näherung \tilde{q}_* auftritt. Die Ergebnisse für die Parameterwerte des neuronalen Netzes dienen schließlich als Initialisierung für die eigentliche Steuerphase, die im nächsten Abschnitt präsentiert wird.

7.4 Steuerphase

Nach Bestimmung einer guten anfänglichen Approximation \tilde{q}_* für die optimale Action-Value-Funktion aus (75) stellt sich die Steuerung der Drehzahl der Pumpe auf Grundlage der Beziehung (76) als verhältnismäßig einfach dar. Die eigentliche Steuerphase startet zum Zeitpunkt

$$t_0^{\text{AI}} := t_0 + \Delta t_{\text{mxmn}}^{\text{AI}} + (K_{\text{tp}}^{\text{AI}} + 1) \cdot J_{\text{tp}}^{\text{AI}} \cdot \Delta t_{\text{tp}}^{\text{AI}} + K_{\text{tp}}^{\text{AI}} \cdot \Delta t_{\text{trans}}^{\text{AI}}$$

in unmittelbaren Anschluss an die Testphase in einem Zustand S_0 . Dieser ist, genau wie alle zukünftigen Zustände S_τ , die das System zu den entsprechenden, äquidistant im zeitlichen Abstand von Δt^{AI} definierten Steuerzeitpunkten $t_\tau^{\text{AI}} := t_0^{\text{AI}} + \tau \cdot \Delta t^{\text{AI}}$ annimmt, nach Gleichungen (77) und (82) zu berechnen. Dabei werden für die Auswertungen der Funktion $L(\cdot)$ wie schon in der Testphase \tilde{a}_{max} und \tilde{a}_{min} aus Gleichung (89) benutzt. Die Ermittlung einer Aktion A_τ , welche über die Relationen

$$\begin{aligned} \tilde{A}_\tau &= \Delta n_{\text{min}}^{\text{AI}} + \frac{1}{2} \cdot (A_\tau + 1) \cdot (\Delta n_{\text{max}}^{\text{AI}} - \Delta n_{\text{min}}^{\text{AI}}) \\ u^{\text{AI}}(t) &:= u^{\text{AI}}(t_\tau^{\text{AI}}) + \tilde{A}_\tau \cdot \frac{t - t_\tau^{\text{AI}}}{\Delta t^{\text{AI}}} \quad \text{für } t \in]t_\tau^{\text{AI}}, t_{\tau+1}^{\text{AI}}] \end{aligned} \tag{98}$$

zu Steueranweisungen für die Drehzahl der Pumpe auf dem Intervall $]t_\tau^{\text{AI}}, t_{\tau+1}^{\text{AI}}]$ übersetzt wird, erfolgt prinzipiell dadurch, dass \tilde{q}_* benutzt wird, um über (76) näherungsweise an den Wert $\pi_*(S_\tau)$ der optimalen Policy zu kommen. Allerdings ist es unvorteilhaft die Relation (76) durchgehend auszunutzen, weil die Funktion \tilde{q}_* als Näherung nicht zwingend die besten Aktionen liefert und weil daher das gelegentliche Ausprobieren anderer Aktionen sowie das weitere Erkunden des Systems auch noch nach der Testphase im eigentlichen Betrieb der Pumpe essenziell dafür ist, um die Näherung weiter verbessern zu können [124, S. 3]. Aus diesem Grund wird in jedem Zeitschritt τ für A_τ mit einer Wahrscheinlichkeit von $\varepsilon_{\text{ctrl}}^{\text{AI}} > 0$ eine rein zufällige Aktion ausgewählt und bloß mit einer Wahrscheinlichkeit von $1 - \varepsilon_{\text{ctrl}}^{\text{AI}}$ die Gleichung (76) zur Berechnung einer möglichst günstigen Aktion ausgenutzt, was auf die Definition

$$A_\tau := \begin{cases} \arg \max_{A \in \mathcal{A}(S_\tau)} \tilde{q}_*(S_\tau, A) & \text{für } X_\varepsilon > \varepsilon_{\text{ctrl}}^{\text{AI}}, \\ X_A & \text{für } X_\varepsilon \leq \varepsilon_{\text{ctrl}}^{\text{AI}}, \end{cases} \quad (99)$$

führt [124, S. 32 (Kasten “A simple bandit algorithm”)]. Dabei seien $X_\varepsilon \sim \mathcal{U}(0, 1)$ bzw. $X_A \sim \mathcal{U}(-1, 1)$ zwei auf dem Intervall $[0, 1]$ bzw. $[-1, 1]$ gleichförmig verteilte Zufallsvariablen.

Weil auch während der eigentlichen Steuerphase alle Parameter $\mathbf{b}^{[l]}$ und $\mathbf{W}^{[l]}$ des neuronalen Netzes sowie \bar{R} mit den Lernraten $\alpha_{\text{ctrl}}^{\text{AI}} > 0$ und $\beta_{\text{ctrl}}^{\text{AI}} > 0$ weitertrainiert werden sollen, ist es notwendig eine Vorschrift für die Berechnung der partiellen Ableitungen von \tilde{q}_* nach den Parametern aufzustellen. Mit den aus Formel (90) bekannten Hilfsgrößen, welche zur Ermittlung des Outputs des neuronalen Netzes eingeführt wurden, ergibt sich diesbezüglich

$$\left. \begin{aligned} \frac{\partial \tilde{q}_*}{\partial \mathbf{z}^{[L]}} &:= \boldsymbol{\sigma}^{[L]'}(\mathbf{z}^{[L]}) \\ \frac{\partial \tilde{q}_*}{\partial \mathbf{W}^{[l]}} &:= \left(\frac{\partial \tilde{q}_*}{\partial \mathbf{z}^{[l]}} \right) (\mathbf{h}^{[l-1]})^T \\ \frac{\partial \tilde{q}_*}{\partial \mathbf{b}^{[l]}} &:= \frac{\partial \tilde{q}_*}{\partial \mathbf{z}^{[l]}} \\ \frac{\partial \tilde{q}_*}{\partial \mathbf{z}^{[l-1]}} &:= \left((\mathbf{W}^{[l]})^T \left(\frac{\partial \tilde{q}_*}{\partial \mathbf{z}^{[l]}} \right) \right) \odot \left(\boldsymbol{\sigma}^{[l-1]'}(\mathbf{z}^{[l-1]}) \right) \\ \frac{\partial \tilde{q}_*}{\partial \mathbf{W}^{[1]}} &:= \left(\frac{\partial \tilde{q}_*}{\partial \mathbf{z}^{[1]}} \right) \mathbf{e}^T \\ \frac{\partial \tilde{q}_*}{\partial \mathbf{b}^{[1]}} &:= \frac{\partial \tilde{q}_*}{\partial \mathbf{z}^{[1]}} \end{aligned} \right\} \quad l \in \{L, \dots, 2\} \quad (100)$$

[103, 4:04–11:13 min]. Damit lässt sich schließlich das sogenannte “Differential semi-gradient Sarsa”-Verfahren aus dem Feld des Reinforcement Learnings implementieren, auf dessen Basis die Drehzahl der Pumpe geregelt wird. Der Algorithmus 2 zeigt eine Übersicht des gesamten Vorgehens.

$\mathbf{b}^{[l]}, \mathbf{W}^{[l]}$ für $l \in \{1, \dots, L\} \leftarrow$ Initialisierung der Parameter des neuronalen Netzes für \tilde{q}_* nach Abschnitt 7.3

$$\bar{R} \leftarrow \bar{r}_{\text{tp}}^{(K_{\text{tp}}^{\text{AI}}, J_{\text{tp}}^{\text{AI}} - 1)}$$

$S_0 \leftarrow$ Initialisierung nach (77) und (82)

$A_0 \leftarrow$ Initialisierung nach (99)

for $\tau \leftarrow 0$ **to** ∞ **do**

 Steuere die Drehzahl der Pumpe auf dem Intervall $[t_\tau, t_{\tau+1}]$ nach

 Formel (98) und führe so die Aktion A_τ aus

 Berechne die Belohnung $R_{\tau+1}$ nach Gleichung (88) und ermittle den neuen Zustand $S_{\tau+1}$ nach Gleichungen (77) und (82)

 Wähle die neue Aktion $A_{\tau+1}$ nach (99)

$$\delta \leftarrow R_{\tau+1} - \bar{R} + \tilde{q}_*(S_{\tau+1}, A_{\tau+1}) - \tilde{q}_*(S_\tau, A_\tau)$$

$$\bar{R} \leftarrow \bar{R} + \beta_{\text{ctrl}}^{\text{AI}} \cdot \delta$$

$$\frac{\partial \tilde{q}_*}{\partial \mathbf{b}^{[l]}} \leftarrow \text{Berechnung gemäß Formeln (90) und (100)} \quad \text{für } l \in \{1, \dots, L\}$$

$$\frac{\partial \tilde{q}_*}{\partial \mathbf{W}^{[l]}} \leftarrow \text{Berechnung gemäß Formeln (90) und (100)} \quad \text{für } l \in \{1, \dots, L\}$$

$$\mathbf{b}^{[l]} \leftarrow \mathbf{b}^{[l]} + \alpha_{\text{ctrl}}^{\text{AI}} \cdot \delta \cdot \frac{\partial \tilde{q}_*}{\partial \mathbf{b}^{[l]}} \quad \text{für } l \in \{1, \dots, L\}$$

$$\mathbf{W}^{[l]} \leftarrow \mathbf{W}^{[l]} + \alpha_{\text{ctrl}}^{\text{AI}} \cdot \delta \cdot \frac{\partial \tilde{q}_*}{\partial \mathbf{W}^{[l]}} \quad \text{für } l \in \{1, \dots, L\}$$

end

Alg. 2: Neue Strategie für die Steuerung der Drehzahl der Pumpe mittels KI [124, S. 251 (Kasten “Differential semi-gradient Sarsa for estimating $\hat{q} \approx q_*$ ”)]

8 Vergleich und kritische Bewertung der Steuerstrategien

Die kritische Bewertung sowie der quantitative Vergleich der in den Kapiteln 6 und 7 vorgestellten Steuerstrategien steht im Mittelpunkt dieses letzten Kapitels. Dabei wird die theoretische Lösung nach dem Optimalitätsprinzip von Bellman aus dem Abschnitt 6.2 als Referenzlösung für alle Vergleiche dienen. Außerdem werden, falls nicht explizit anders angegeben, für die Bestimmung aller Ergebnisse dieses Kapitels die im Anhang C angegebenen Werte der Parametrierung des DGL-Systems sowie die der Parametrierung der einzelnen Steuerstrategien verwendet.

Die unterschiedlichen Verfahren möglichst fair miteinander vergleichen zu können, stellt sich speziell für die erweiterten Steuerstrategien und für die neue Steuerstrategie mittels KI aus diversen Gründen als prinzipiell nicht so leicht heraus. Das liegt zum Beispiel daran, dass beim KI Ansatz und dem von Grundfos Testphasen eine Rolle spielen, wohingegen dies bei den elementaren Strategien und dem Verfahren von KSB nicht der Fall ist. Demgegenüber hängen die Resultate beim Ansatz von KSB stark von der Wahl der statischen Regelkurve ab, während weder bei Grundfos noch beim KI Ansatz überhaupt eine statische Regelkurve zum Einsatz kommt. Eine weitere Herausforderung für die Durchführung von Vergleichen ergibt sich aufgrund der Nebenbedingung G des optimalen Steuerproblems (50), weil für praxistaugliche Werte von T_{tol} mit kurzzeitigen Verletzungen der Nebenbedingung zu rechnen ist, da die Steuerstrategien unter Umständen nicht schnell genug auf die Veränderung einer Störgröße des Systems reagieren können, etwa wenn die Außentemperatur schlagartig stark abfällt. Um der letztgenannten Problematik entgegenzuwirken, wird im Folgenden als Alternative zu $G(\mathbf{y}(t), u(t))$ auch die Funktion

$$G_2(\mathbf{y}(t), u(t)) := \frac{1}{\lambda([t_0, t_f] \setminus U_{\text{test}})} \cdot \int_{\substack{t \in [t_0, t_f] \\ t \notin U_{\text{test}}}} \max(T_{\text{set}}(t) - T_{\text{room}}(t), 0.0 \text{ K}) dt - T_{\text{tol}}$$

hinzugezogen, mit der sich bewerten lässt, ob die Raumtemperatur durch die jeweils gewählte Steuerung im Rahmen der vorgegebenen Temperaturtoleranz den Sollwert erreicht. Die Funktion $G_2(\mathbf{y}(t), u(t))$ beschreibt dabei mithilfe des Lebesgue-Maßes $\lambda(\cdot)$ [7, S. 19 f. (Kap. 3)] den Mittelwert des Positivteils der Differenz zwischen der Soll- und Raumtemperatur bezogen auf den gesamten Zeit-

raum $[t_0, t_f]$ mit Ausnahme der Vereinigung aller zeitlichen Intervalle U_{test} , in denen gegebenenfalls ein Drehzahltest stattfindet.

Für die Durchführung von Vergleichen ist es nützlich zunächst eine Referenz zu ermitteln, wobei die Lösung, die sich aus dem Optimalitätsprinzip von Bellman ergibt, dafür prädestiniert ist. Die Konstruktion nach Abschnitt 6.2 stellt nämlich einerseits die beste Näherung für die Lösung des Steuerproblems (50) dar, ist andererseits aber für praktische Zwecke außer Konkurrenz. Dies begründet sich damit, dass die optimalen Werte der Parameter u_j nur dann bestimmt werden können, wenn die Antworten des Systems für unterschiedliche Werte von u_j untereinander vergleichbar sind, also vorausgesetzt wird, dass die Reaktion des Systems auf eine von u_j abhängige Drehzahländerung bereits vor Durchführung der Änderung bekannt ist. Diese Voraussetzung ist jedoch nur in der Theorie und nicht in der Praxis erfüllbar. Die Abbildung 8.1 zeigt den Drehzahlverlauf, welcher sich gemäß der nach Bellman konstruierten Lösung $\tilde{u}_*^{\text{Bm}}(t)$ aus Gleichung (56) ergibt. Dabei bleibt die Nebenbedingung des optimalen Steuerproblems unter Berücksichtigung der Toleranz T_{tol} mit $G \approx -1.0 \cdot 10^{-8}$ K erfüllt und die Zielfunktion liefert den Referenzwert $J_{\text{ref}} \approx 2414.1$ Wh.

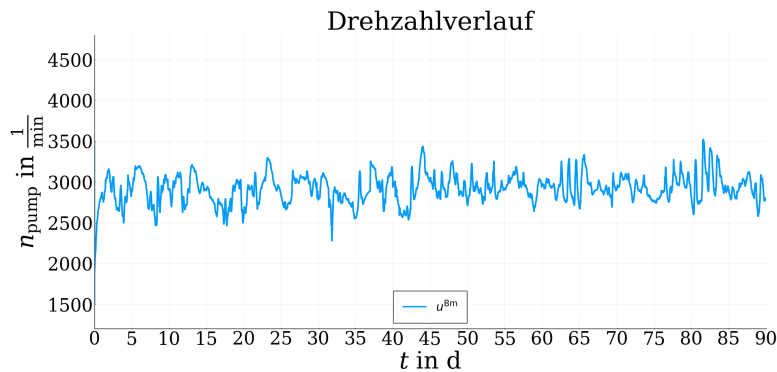


Abb. 8.1: Drehzahlverlauf für die Lösung nach Bellman

Nach Berechnung des Referenzwertes werden nun als Erstes die vier elementaren Steuerstrategien aus Abschnitt 6.3 untersucht und miteinander verglichen. Hierbei fungiert der Auslegungspunkt $(\dot{V}_{\text{dsgn}}, H_{\text{dsgn}})$, der sich aus einem idealen hydraulischen Abgleich des Heizsystems des Benchmark-Modells ergeben würde, als Orientierungspunkt für die Wahl der Parameter der Steuerstrategien. Weil der optimale Auslegungspunkt in der Praxis nur mit großem Aufwand bestimmbar ist [35] und weil ohnehin dazu tendiert wird für die Parameter einer Steuerstrategie einen Sicherheitsaufschlag hinzuzurechnen, werden an dieser Stelle die Werte so ausgewählt, dass für ein $o_{\text{dsgn}} > 0$ die Regelkurven der elementaren Strategien alle durch den Punkt $(1 + o_{\text{dsgn}}) \cdot (\dot{V}_{\text{dsgn}}, H_{\text{dsgn}})$ verlaufen. In der Abbildung 8.2 sind die jeweiligen Drehzahlverläufe zu sehen,

8 Vergleich und kritische Bewertung der Steuerstrategien

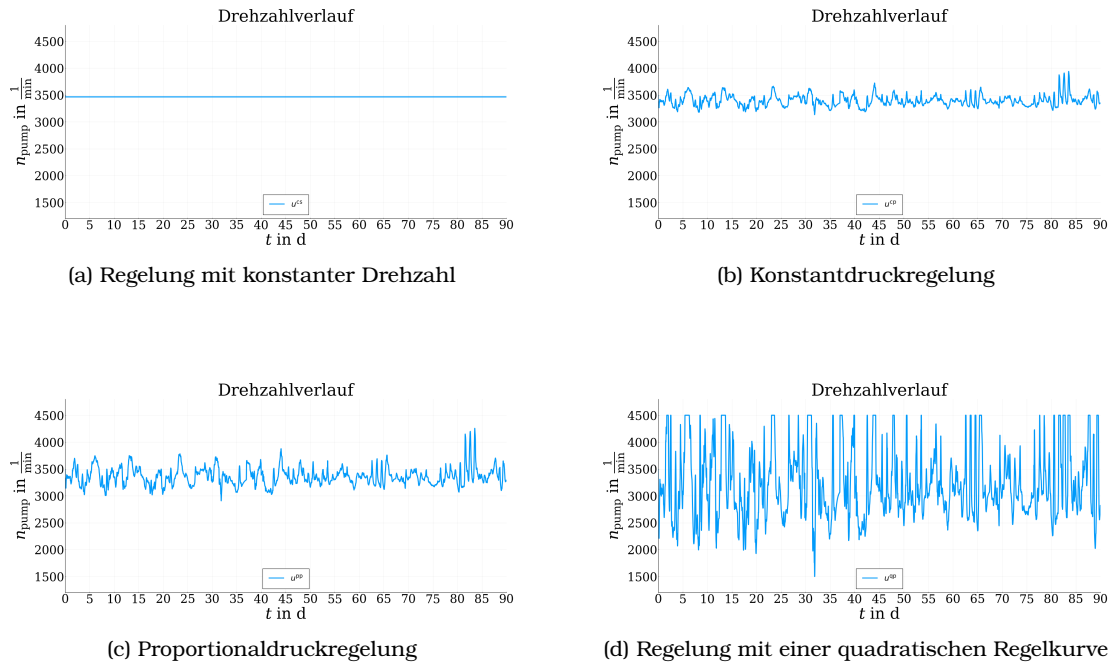


Abb. 8.2: Drehzahlverläufe für die elementaren Steuerstrategien

Regelung	$J / J_{\text{ref}} - 1$	G_2
u^{CS}	64.7 %	-0.222 K
u^{CP}	56.6 %	-0.202 K
u^{PP}	53.1 %	-0.191 K
u^{QP}	50.7 %	-0.110 K

Tab. 8.1: Vergleich des Integrals über die hydraulische Leistung mit dem Referenzwert J_{ref} für die vier elementaren Steuerstrategien

die daraus resultieren, und die Tabelle 8.1 listet die jeweiligen relativen Abweichungen des Integrals über die hydraulische Leistung zum Referenzwert J_{ref} auf. Die Ergebnisse entsprechen den Erwartungen, weil eine Proportionaldruckregelung oder eine Regelung mit einer quadratischen Regelkennlinie im allgemeinen Fall energetisch günstiger ist als eine Konstantdruckregelung und diese ihrerseits wiederum besser ist als eine unregelte Pumpe, die nur mit einer konstanten Drehzahl betrieben wird [87].

Bezüglich der Steuerstrategie von KSB ist es naheliegend, die Ergebnisse der reinen, grundlegenden Steuerung $u_{\text{stat}}^{\text{KSB}}$ ohne den von KSB entwickelten dynamischen Anteil $u_{\text{dyn}}^{\text{KSB}}$ mit den Ergebnissen zu vergleichen, die sich ergeben, wenn der dynamische Anteil Berücksichtigung findet und gemäß der von KSB entwickelten Kriterien zu geeigneten Zeitpunkten nach Gleichung (65) zu einer

8 Vergleich und kritische Bewertung der Steuerstrategien

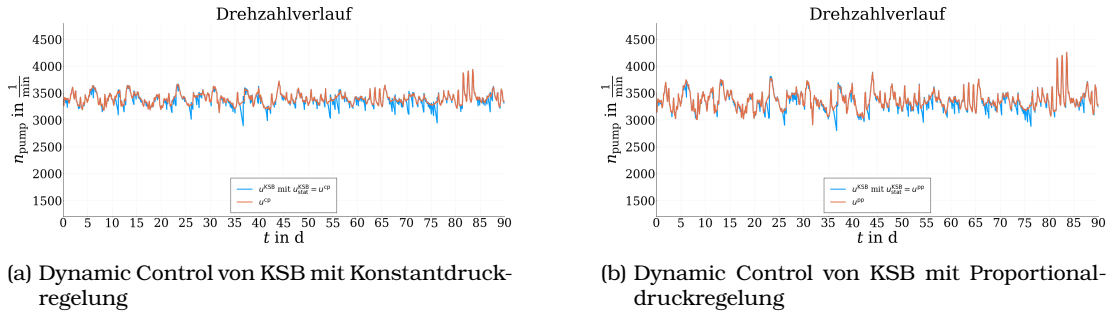


Abb. 8.3: Drehzahlverlauf für die Steuerstrategie Dynamic Control von KSB

Regelung	$J / J_{\text{ref}} - 1$	G_2
$u^{\text{CP}}, o_{\text{dsgr}} > 0$	56.6 %	-0.202 K
u^{KSB} mit $u_{\text{stat}}^{\text{KSB}} = u^{\text{CP}}, o_{\text{dsgr}} > 0$	54.0 %	-0.194 K
$u^{\text{CP}}, o_{\text{dsgr}} = 0$	33.4 %	-0.137 K
u^{KSB} mit $u_{\text{stat}}^{\text{KSB}} = u^{\text{CP}}, o_{\text{dsgr}} = 0$	30.4 %	-0.126 K
$u^{\text{PP}}, o_{\text{dsgr}} > 0$	53.1 %	-0.191 K
u^{KSB} mit $u_{\text{stat}}^{\text{KSB}} = u^{\text{PP}}, o_{\text{dsgr}} > 0$	50.8 %	-0.183 K
$u^{\text{PP}}, o_{\text{dsgr}} = 0$	37.4 %	-0.148 K
u^{KSB} mit $u_{\text{stat}}^{\text{KSB}} = u^{\text{PP}}, o_{\text{dsgr}} = 0$	35.1 %	-0.138 K

Tab. 8.2: Vergleich des Integrals über die hydraulische Leistung mit dem Referenzwert J_{ref} für die Steuerstrategie Dynamic Control von KSB

Drehzahlabsenkung führt. Zu diesem Zweck werden hier einerseits sowohl die Konstantdruckregelung als auch die Proportionaldruckregelung als elementare Steuerungen mit den gleichen Parametern, die schon im Absatz zuvor gewählt wurden, hinterlegt, andererseits aber auch die jeweiligen Regelungen mit $o_{\text{dsgr}} = 0$ betrachtet. Die Abbildung 8.3, welche die Drehzahlverläufe für die Fälle mit $o_{\text{dsgr}} > 0$ darstellt, sowie die Tabelle 8.2, welche den Vergleich des Integrals über die hydraulische Leistung mit dem Referenzwert J_{ref} beinhaltet, gehen daraus hervor. Es zeigt sich, dass mit dem Ansatz von KSB stets etwa zwischen vier und acht Prozent eingespart werden können. Mit Blick auf die Tabelle wird aber ebenso klar, dass die Wahl der statischen Steuerung $u_{\text{stat}}^{\text{KSB}}$ einen größeren und viel entscheidenderen Einfluss auf das Endergebnis hat.

Sowohl beim Verfahren Autoadapt von Grundfos als auch bei der neuen Steuerstrategie mittels KI spielen Drehzahltests eine entscheidende Rolle, sodass hier die relativen Abweichungen zu J_{ref} auch die unter Umständen ungünstigen Testphasen beinhalten. Speziell für die neue Steuerstrategie, bei der der eigentlichen Steuerphase eine Testphase vorausgeht, ist der Einfluss der Testphase auf J aber abhängig von dem betrachteten Endzeitpunkt t_f . Um die Aussagefähigkeit und die Vergleichbarkeit der Resultate mit den anderen Stra-

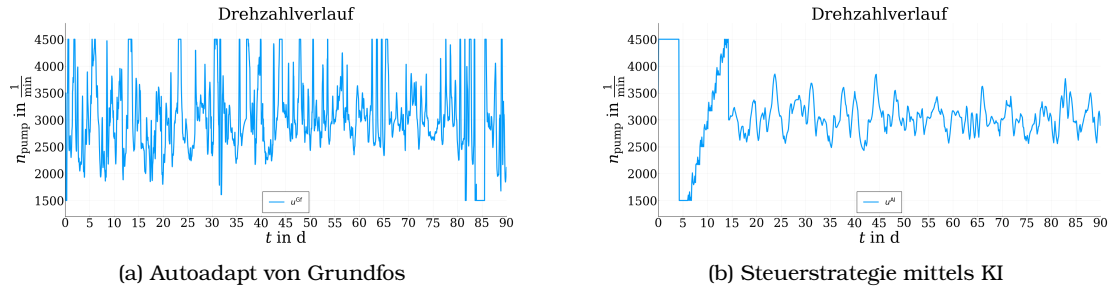


Abb. 8.4: Drehzahlverlauf für die Steuerstrategie Autoadapt von Grundfos und für die neue Steuerstrategie mittels KI

Regelung	$J / J_{\text{ref}} - 1$	G_2
u^{Gf}	28.5 %	-0.003 K
u^{AI}	25.8 %	-0.025 K

Tab. 8.3: Vergleich des Integrals über die hydraulische Leistung mit dem Referenzwert J_{ref} für die Steuerstrategie Autoadapt von Grundfos und für die neue Steuerstrategie mittels KI

tegien für diese zwei Verfahren zu erhöhen, werden deshalb neben den relativen Abweichungen von J zu J_{ref} zusätzlich noch relative Abweichungen von \hat{J} zu \hat{J}_{ref} ermittelt. Dabei bezeichnet

$$\hat{J}(\mathbf{y}(t), u(t)) := \rho_{\text{wg}} \cdot \int_{\substack{t \in [t_0, t_f] \\ t \notin U_{\text{test}}}} H(t) \cdot \dot{V}(t) dt$$

den vom jeweiligen Verfahren abhängigen Wert des Integrals über die hydraulische Leistung bei Auslassung der Vereinigung aller zeitlichen Intervalle U_{test} , in denen ein Drehzahltest durchgeführt wird, und \hat{J}_{ref} repräsentiert den entsprechenden Wert für die mit dem Prinzip von Bellman berechnete Lösung. In der Abbildung 8.4 finden sich die Drehzahlverläufe wieder, die sich bei Anwendung des Autoadapt Algorithmus von Grundfos und bei Anwendung der neuen Strategie ergeben. Außerdem zeigen die Tabellen 8.3 beziehungsweise 8.4 die relativen Abweichungen von J zu J_{ref} beziehungsweise von \hat{J} zu \hat{J}_{ref} . Aus den Simulationsergebnissen geht zwar quantitativ hervor, dass die neue Steuerstrategie mittels KI ein wenig besser abschneidet als Grundfos' Autoadapt, dennoch kann die Schlussfolgerung daraus nicht sein, dass dies auch zwingend für die Praxis gelten muss. Die Tatsache, dass der Autor dieser Arbeit das Verfahren von Grundfos lediglich auf Basis der wenigen, öffentlich bekannten Informationen nach seinen eigenen Vorstellungen implementiert hat, führt dazu, dass die hier präsentierten Ergebnisse für den Algorithmus von Grundfos als eine

Regelung	$\hat{J} / \hat{J}_{\text{ref}} - 1$
u^{Gf}	29.8 %
u^{AI}	14.9 %

Tab. 8.4: Vergleich des Integrals über die hydraulische Leistung mit dem Referenzwert \hat{J}_{ref} für die Steuerstrategie Autoadapt von Grundfos und für die neue Steuerstrategie mittels KI

obere Schranke für den unbekanntem, tatsächlichen Autoadapt Algorithmus angesehen werden müssen. Diese Interpretation der Resultate zugrunde legend erlaubt es also nicht zu entscheiden, welches dieser zwei Verfahren für das Benchmark-Problem besser geeignet ist. Allerdings lassen die Ergebnisse bei Vernachlässigung der Testphasen zumindest die Vermutung zu, dass beide Verfahren in puncto Energieeinsparung ähnlich gut sind.

Für eine kritische Bewertung aller Verfahren ist jedoch wichtig nicht nur das Energieeinsparpotential, sondern auch die Robustheit zu betrachten. Hier ist das Verfahren von KSB, das energetisch nur an die Ergebnisse der Grundfos und KI Strategie herankommt, wenn eine günstige statische Regelkurve $u_{\text{stat}}^{\text{KSB}}$ hinterlegbar ist, im Vorteil. Dadurch, dass die dynamische Steuerung nämlich in Störfällen nach Gleichung (64) für eine Dauer von mindestens Δt^{KSB} auf $u_{\text{stat}}^{\text{KSB}}$ zurückgesetzt wird, ist das Verfahren von KSB sehr robust [31, S. 2]. Demgegenüber sind die Grundfos- und KI-Strategie aufgrund ihrer Drehzahltests stör anfälliger. So lieferte der Autoadapt Algorithmus bei einem Feldtest in etwa drei von vier Fällen eine Verbesserung gegenüber ihrer bisherigen Standardregelung, führte jedoch auch in ungefähr 11 % zu einer Verschlechterung [13, S. 29f.]. Obwohl die neue Strategie des Autors noch keinem Feldtest unterzogen wurde, ist nicht zu erwarten, dass Störungen keinen Einfluss auf die Regelung haben. Insbesondere die anfängliche Testphase macht das Verfahren anfällig, weil die eigentliche Steuerphase auf eine gute Näherung von q_* angewiesen ist. Wenn die Testphase beispielsweise bei recht hohen Außentemperaturen durchgeführt würde, würden die Ventile aller Voraussicht nach selbst bei der niedrigsten Drehzahl n_{min} nicht ganz öffnen. Infolgedessen würde ein ungünstiger Wert für \tilde{a}_{min} aus Formel (89) ermittelt, sodass die Steuerung nicht wie erwartet und gewünscht laufen könnte. Wenn jedoch vor Beginn der eigentlichen Steuerphase mit den in der Testphase gesammelten Daten eine gute Approximation \tilde{q}_* berechnet werden kann, ist davon auszugehen, dass einzelne, kurzzeitige Störungen etwa von außen oder durch den Nutzer, nur zu kurzzeitig negativen Effekten führen und die Steuerung somit robust ist. Anders ausgedrückt ist die KI Steuerstrategie also stark davon abhängig, dass in der Testphase der Einfluss von Störungen minimal ist, kann anschließend aber stabil laufen, was ein Vorteil gegenüber dem Verfahren von Grundfos ist, bei dem immer wieder Drehzahltests mit n_{min} oder n_{max} durchgeführt werden.

9 Fazit und Ausblick

Auch wenn in typischen Einfamilienhäusern der Anteil der verbrauchten Heizenergie deutlich höher ist als der Anteil der verbrauchten Energie der verbauten Pumpe [144, S. 3], bemühen sich Hersteller, ihre Steueralgorithmen für Umwälzpumpen weiterzuentwickeln. Denn selbst Lösungen, die lediglich zu kleinen Energieeinsparungen führen, stellen nicht nur ein besseres Verkaufsargument dar, sondern können auch über die Vorbildfunktion gesellschaftlichen Einfluss auf den Umweltschutz nehmen [135, S. 15]. Zudem besitzen gute, robuste und in der Praxis erprobte Regelungen das Potenzial, als Vorlage für Algorithmen wesentlich größerer Pumpensysteme zu dienen, in denen sich dann quantitativ mehr Energie einsparen lässt.

Für die Entwicklung von Steuerstrategien von Heizungspumpen wird häufig ein Einraum-Heizsystem als Benchmark-Modell für Simulationen benutzt. Ein umfangreiches Modell, das dafür bestens geeignet ist, hat der Autor dieser Arbeit mit der Programmiersprache Julia auf Basis des Dymola-Modells von Webersinke [134] und Kowalski [75] umgesetzt und vorgestellt. Insbesondere konnte gezeigt werden, dass das mit Julia implementierte Benchmark-Modell der mit Modelica in Dymola implementierten Variante bezüglich der CPU-Zeit, die für die Berechnung aller Simulationsergebnisse benötigt wird, um einen Faktor von fast 50 schneller ist.

Des Weiteren ist untersucht worden, ob die als unbekannt angenommenen Parameter des Benchmark-Modells nur auf Basis der Daten, die der Pumpe während ihres Betriebes zur Verfügung stehen, identifizierbar sind. Hierbei stellte sich heraus, dass dies theoretisch zwar möglich ist, der Rechenaufwand dafür jedoch unter realistischen und praktischen Bedingungen zu hoch ist. Daraus ergab sich die Schlussfolgerung, dass für die Entwicklung einer neuen Steuerstrategie der Pumpe in einem Heizsystem modellfreie Ansätze modellbasierten vorzuziehen sind.

Auf dieser Erkenntnis aufbauend hat der Autor der Arbeit mithilfe eines modellfreien Reinforcement Learning Ansatzes eine neue Strategie für die Steuerung der Pumpe entworfen. Die Grundidee besteht dabei darin mit einer Testphase zu starten, in der bewusst viele verschiedene Systemzustände erzeugt werden, um hinreichend viele Daten zu sammeln, die das Systemverhalten abbilden können. Mit den gesammelten Daten ist es dann möglich ein neuronales Netz als Näherung für die Action-Value-Funktion des Reinforcement Learning Ansatzes vorzutrainieren. Diese Näherung wiederum, die während des Pumpenbetriebs durch Auswertung der neu hinzukommenden Daten fortlaufend

weiter verbessert wird, bietet die Gelegenheit auf relativ einfache Weise Drehzahländerungen für die Pumpe zu ermitteln, die günstig im Sinne eines zuvor definierten Optimierungsproblems für das Benchmark-Modell sind.

Neben seiner eigens neu entwickelten Strategie wurden für die Durchführung von Vergleichen vom Autor auch wichtige existierende Steuerstrategien präsentiert. Angefangen von den elementaren, leicht umsetzbaren Verfahren über eine neuere Strategie namens Dynamic Control des Pumpenherstellers KSB bis hin zu dem seit Langem existierenden besten Algorithmus Autoadapt von Grundfos. Die exakten quantitativen Resultate, die sich aus den Vergleichen ergeben, sind aus diversen Gründen mit Vorsicht zu betrachten, weil es schwierig ist den Einfluss von unterschiedlich durchgeführten Testphasen einander gegenüberzustellen. Außerdem stellt die Implementierung des Autoadapt Verfahrens nicht die Originalversion von Grundfos dar Nichtsdestotrotz lassen sich aus den Resultaten qualitativ einige wichtige Erkenntnisse gewinnen. Das Verfahren Dynamic Control von KSB ist eine kleine Verbesserung im Vergleich zu den jeweils zugrunde liegenden elementaren Steuerstrategien, aber seine Güte steht und fällt mit der Wahl der elementaren Steuerstrategie und den zugehörigen Parametern. Nur für eine sehr gute Wahl kann das Verfahren mit der neuen Steuerstrategie des Autors oder dem Autoadapt Verfahren von Grundfos mithalten. Dafür ist es jedoch sehr robust. Die neue Steuerstrategie des Autors hingegen bewegt sich aufgrund ihrer selbstlernenden Charakteristik auf dem Niveau des Autoadapt Algorithmus. Die Frage, ob dieses neue, auf einem Reinforcement Learning Ansatz beruhende Verfahren, so wie die Simulationsergebnisse es andeuten, sogar den Autoadapt Algorithmus in der Praxis schlagen könnte, kann hier nicht abschließend beantworten werden. Im Gegensatz zu dem Verfahren von Grundfos, was sich seit Jahren in der Praxis bewährt hat, ist dieses nämlich noch nicht in einem physischen System implementiert und getestet worden.

In Bezug darauf ist es fraglich, ob es für irgendeinen Pumpenhersteller überhaupt noch als lohnend erscheint die Idee genauso wie vorgeschlagen umzusetzen. Denn eine Grundannahme der Pumpenhersteller, nach welcher die Pumpe nicht mit den restlichen Elementen des Heizsystems kostengünstig und ohne großen Aufwand vernetzt werden kann [46, S. 2 (Sp. 1 f., Abs. 4)], dürfte in Anbetracht der steigenden Smart Home Popularität [109] mehr und mehr in Zweifel gezogen werden. Doch selbst wenn, anders als in dem eingangs erwähnten Forschungsprojekt mit Lowara und dem SOPHi-Projekt mit KSB, in denen die Entwicklung der neuen, hier vorgestellten Steuerstrategie seinen Anfang nahm, auf diese Annahme verzichtet würde und der Pumpe viel mehr Daten des Systems zur Verfügung stünden, könnte eine modifizierte Version des Reinforcement Learning Ansatzes des Autors genutzt werden. Es bleibt jedoch abzuwarten, wie es mit dem Smart Home Trend weitergeht und wie sich die Pumpenhersteller dem anpassen werden.

Glossar

Anlagenkennlinie Kurve im \dot{V} - Δp -Diagramm, die in guter Näherung eine Parabel ist und für einen vorgegeben Volumenstrom angibt, wie groß die gesamte von der Pumpe zu überwindende Druckdifferenz über alle hydraulischen Widerstände des Heizsystems ist.

Autoadapt Die vom Pumpenhersteller Grundfos entwickelte Strategie für die Steuerung der Drehzahl einer Umwälzpumpe in einem Heizsystem.

Außentemperatur Die Temperatur, welche die Luft außerhalb des Gebäudes, das beheizt wird, annimmt.

Benchmark-Modell Modell eines Einraum-Heizsystems, das im Wesentlichen aus einer drehzahlgesteuerten Pumpe, einem Kessel, Rohren, einem Heizkörper mit Thermostatventil, Boden, Decke, Wänden und Fenstern sowie einem Nutzer besteht.

Benchmark-Problem Optimales Steuerproblem für die Drehzahl der Pumpe des Benchmark-Modells.

Boden- und Deckentemperatur Die Temperatur, welche der Boden und die Decke eines Raumes des beheizten Gebäudes zusammen annehmen.

Dynamic Control Die vom Pumpenhersteller KSB entwickelte Strategie für die Steuerung der Drehzahl einer Umwälzpumpe in einem Heizsystem.

Förderhöhe Eine zur Druckdifferenz proportionale Größe, welche in Heizsystemen häufig anstelle der Druckdifferenz betrachtet wird.

Heizkörpertemperatur Die Temperatur, welche der Heizkörper eines Raumes des beheizten Gebäudes annimmt.

Konstantdruckregelung Eine elementare Strategie für die Steuerung der Drehzahl einer Pumpe, bei der die Drehzahl stets so angepasst wird, dass der Differenzdruck Δp des Betriebspunkts der Pumpe einen vorgegebenen Wert hält.

Proportionaldruckregelung Eine elementare Strategie für die Steuerung der Drehzahl einer Pumpe, bei der die Drehzahl stets so angepasst wird, dass der Betriebspunkt der Pumpe sich auf einer vorgegebenen linearen Gerade im \dot{V} - Δp -Kennfeld befindet.

- Pumpenkennlinie** Kurve im \dot{V} - Δp -Diagramm, die angibt, welche Druckdifferenz die Pumpe bei einem vorgegebenen Volumenstrom überwinden kann.
- Raumtemperatur** Die Temperatur, welche die Luft innerhalb eines Raumes des beheizten Gebäudes, annimmt.
- Regelung mit konstanter Drehzahl** Eine elementare Strategie für die Steuerung der Drehzahl einer Pumpe, bei der die Pumpe mit einer vorgegebenen, konstanten Drehzahl betrieben wird.
- Regelung mit quadratischer Kennlinie** Eine elementare Strategie für die Steuerung der Drehzahl einer Pumpe, bei der die Drehzahl stets so angepasst wird, dass der Betriebspunkt der Pumpe sich auf einer vorgegebenen Parabel im \dot{V} - Δp -Kennfeld befindet.
- Rücklauftemperatur** Die nach einem Durchlauf durch das Heizsystem abgekühlte Temperatur des Wassers, das dem Heizkessel zum erneuten Erhitzen wieder zugeführt wird.
- Ventilkennlinie** Kurve, die die Abhängigkeit des Ventilöffnungsgrads von der Differenz der Wunschttemperatur des Nutzers und der aktuellen Raumtemperatur beschreibt.
- Ventilöffnungsgrad** Prozentuales Verhältnis zwischen der Größe der Fläche, die beim gegenwärtigen Zustand des Ventils durchflossen wird und der maximalen Fläche, die beim vollständig geöffnetem Ventil durchflossen werden kann.
- Volumenstrom** Der Wasserdurchfluss pro Zeiteinheit, mit dem das vom Heizkessel erwärmte Wasser zu den Heizkörpern gepumpt wird.
- Vorlauftemperatur** Die vom Heizkessel erzeugte Temperatur des Wassers, das durch die Pumpe dem Heizsystem zugeführt wird.
- Wandtemperatur** Die Temperatur, welche die Wand eines Raumes des beheizten Gebäudes annimmt.
- Wunschttemperatur des Nutzers** Wunschttemperatur für einen Raum, die vom Nutzer aktiv durch das Ändern der Stellung des Thermostatkopfs (Skala 0–5) beeinflusst werden kann.

Symbolverzeichnis

\times	Operator für das Kreuzprodukt zweier Vektoren
$\dot{\circ}$	Ein Punkt über einer Größe kann sowohl für die totale Zeitableitung dieser Größe stehen als auch kennzeichnen, dass es sich um eine Flussgröße handelt
\setminus	Operator für die Differenz zweier Mengen
\oplus	Operator für die spaltenweise Addition eines Vektors zu jeder Spalte einer Matrix
\otimes	Operator für das dyadische Produkt zweier Vektoren
\odot	Operator für die komponentenweise Multiplikation zweier Matrizen
$\mathbf{0}$	Nullvektor
A	Oberfläche eines Körpers
\mathcal{A}	Die Ausgangskonfiguration aller materiellen Punkte eines Körpers
$\mathcal{A}(s)$	Menge aller möglichen Aktionen, die ein Agent im Zustand s ausführen kann
$\tilde{\mathcal{A}}_\tau(S_\tau)$	Menge aller möglichen Drehzahländerungen, die während eines Steuerschritts bei der neuen Steuerstrategie mittels KI vorgenommen werden können, wenn das System sich im Zustand S_τ befindet
A_{pipe}	Querschnittsfläche eines Rohres
A_{wall}	Effektive Wandfläche
A_{win}	Fläche aller Fenster in einer Wand
A_τ	Aktion, die der Agent bei der Steuerung mittels KI im τ -ten Zeitschritt ausführt
\tilde{A}_τ	Drehzahländerung, die während eines Steuerschritts bei der neuen Steuerstrategie mittels KI vorgenommen wird
$\mathcal{C}(\cdot, \dots, \cdot)$	Mittlere quadratische Fehlerverlustfunktion, die für das Training des neuronalen Netzes bei der Initialisierung der Steuerphase der neuen Steuerstrategie mittels KI eingesetzt wird
$C_{\text{tp}}^{\text{AI}}$	Matrix mit ganzzahligen Koeffizienten aus der Menge $\{-K_{\text{tp}}^{\text{AI}}, \dots, K_{\text{tp}}^{\text{AI}}\}$, die während der Haupttestphase der neuen Steuerstrategie mittels KI bei der Definition der Testdrehzahlen $n_{\text{test}}^{(k,j)}$ benutzt wird

E	Matrix aller Eingangsvektoren $e^{(i)}$ des Testdatensatzes, der bei der Initialisierung der Steuerphase der neuen Steuerstrategie mittels KI erstellt wird
E_{kin}	Kinetische Energie
$\mathbb{E}_{\pi}(\cdot \cdot)$	Erwartungswert bezüglich der Policy π bei der Reinforcement Learning Formulierung
F	Deformationsgradient der Transformation von der Ausgangskonfiguration zur aktuellen Konfiguration aller materiellen Punkte eines Körpers
F_{res}	Resultierenden Kraft aller äußeren Kräfte, die auf den Körper wirken
G	Nebenbedingung des optimalen Steuerproblems
G^{KSB}	Übertragungsfunktion, die in der signalorientierten Implementierung der Steuerstrategie Dynamic Control von KSB verwendet wird
G_1^{id}	Vektor mit den Anpassungsbedingungen des Optimierungsproblems zur Parameteridentifikation des Benchmark-Modells
G_2^{id}	Vektor mit den algebraischen Nebenbedingungen des Optimierungsproblems zur Parameteridentifikation des Benchmark-Modells
G_2	Alternative zu der Nebenbedingung G des optimalen Steuerproblems
G_{max}	Oberer Grenzpunkt auf der Regelkurve K_{mid} bei der Steuerstrategie Autoadapt von Grundfos
G_{min}	Unterer Grenzpunkt auf der Regelkurve K_{mid} bei der Steuerstrategie Autoadapt von Grundfos
G_{τ}	Zukünftiger Gewinn bei der Reinforcement Learning Formulierung
H	Förderhöhe
$H^{[l]}$	Für die Batch Formulierung benutzte Matrix der Aktivierungen der l -ten Schicht des neuronalen Netzes für die Näherung der optimalen Action-Value-Funktion bei der neuen Steuerstrategie mittels KI
H^{cp}	Feste Förderhöhe bei einer Konstantdruckregelung
H^{pp}	Ordinatenabschnitt einer linearen Regelkurve
H^{qp}	Vertikale Verschiebung einer quadratischen Regelkurve
H_0	Vertikale Verschiebung der minimalen Anlagenkennlinie K_{min} , der maximalen Anlagenkennlinie K_{max} und der Regelkurve K_{mid} bei der Steuerstrategie Autoadapt von Grundfos
$H_{G_{\text{max}}}$	Förderhöhe des oberen Grenzpunktes G_{max} auf der Regelkurve K_{mid} bei der Steuerstrategie Autoadapt von Grundfos

$H_{G_{\min}}$	Förderhöhe des unteren Grenzpunktes G_{\min} auf der Regelkurve K_{mid} bei der Steuerstrategie Autoadapt von Grundfos
H_{S_1}	Förderhöhe des Hilfspunktes S_1 für die geometrische Interpretation bei der Steuerstrategie Autoadapt von Grundfos
H_{S_2}	Förderhöhe des Hilfspunktes S_2 für die geometrische Interpretation bei der Steuerstrategie Autoadapt von Grundfos
$H_{a_{\max}}$	Förderhöhe des Schnittpunkts zwischen der maximalen Pumpenkennlinie und der maximalen Anlagenkennlinie bei der neuen Steuerstrategie mittels KI
$H_{a_{\min}}$	Förderhöhe des Schnittpunkts zwischen der maximalen Pumpenkennlinie und der minimalen Anlagenkennlinie bei der neuen Steuerstrategie mittels KI
$H_{a_{\text{sys}}}$	Förderhöhe des Schnittpunkts zwischen der maximalen Pumpenkennlinie und der aktuellen Anlagenkennlinie bei der neuen Steuerstrategie mittels KI
H_{dsgn}	Förderhöhe des Auslegungspunktes, welcher sich bei einem idealen hydraulischen Abgleich des Heizsystems ergeben würde
H_n	Eine Förderhöhe der Pumpenkennlinie bei einer festen Drehzahl n
H_{pc}	Hilfsgröße bei der Berechnung von g_{opp}
H_{ref}	Referenzförderhöhe
H_{sys}	Eine Förderhöhe auf der Anlagenkennlinie
$H_{\text{sys},0}$	Vertikale Verschiebung der parabelförmigen Anlagenkennlinie
I_{all}	Indexmenge für alle Zustandsgrößen des Benchmark-Modells
I_{msr}	Indexmenge für alle messbaren Zustandsgrößen des Benchmark-Modells
J	Zielfunktion des optimalen Steuerproblems, welche den Wert des Integrals über die hydraulische Leistung verkörpert
\hat{J}	Integral über die hydraulische Leistung, bei dem alle Zeitintervalle, zu denen Drehzahltests durchgeführt werden, unberücksichtigt bleiben
$J_{\text{tp}}^{\text{AI}}$	Anzahl der Drehzahltests, die für jede Bezugsdrehzahl während der Haupttestphase der neuen Steuerstrategie mittels KI durchgeführt wird
\mathbf{J}^{id}	Vektorielle Zielfunktion des Optimierungsproblems zur Parameteridentifikation des Benchmark-Modells
J_{ref}	Als Referenzwert für Vergleiche dienender Wert der Zielfunktion J des optimalen Steuerproblems für die Lösung, die sich aus dem Optimalitätsprinzip von Bellman ergibt
\hat{J}_{ref}	Als Referenzwert für Vergleiche dienender Wert von \hat{J} für die Lösung, die sich aus dem Optimalitätsprinzip von Bellman ergibt

\mathcal{K}	Bezeichnung für einen Körper, der sich im Sinne der Kontinuumsmechanik aus einer Menge von materiellen Punkten zusammensetzt
$K_{\text{tp}}^{\text{AI}}$	Maximaler Index für die Anzahl der Bezugsdrehzahlen während der Haupttestphase der neuen Steuerstrategie mittels KI
K^{id}	Anzahl der Intervalle, die beim Mehrfachschießverfahren genutzt werden
K_{max}	Aktuell maximale Anlagenkennlinie bei der Steuerstrategie Autoadapt von Grundfos
K_{mid}	Quadratische Regelkennlinie der Steuerstrategie Autoadapt von Grundfos
K_{min}	Aktuell minimale Anlagenkennlinie bei der Steuerstrategie Autoadapt von Grundfos
L	Anzahl der Schichten (ohne Eingabeschicht) des neuronalen Netzes für die Näherung der optimalen Action-Value-Funktion bei der neuen Steuerstrategie mittels KI
$L(\cdot)$	Hilfsfunktion für die Berechnung von $\tilde{\alpha}$ bei der neuen Steuerstrategie mittels KI
$\mathcal{L}(\cdot)$	Laplace-Transformation
$\mathcal{L}(\cdot, \cdot)$	Quadratische Fehlerverlustfunktion, die für das Training des neuronalen Netzes bei der Initialisierung der Steuerphase der neuen Steuerstrategie mittels KI eingesetzt wird
$L_{\mathcal{O}}$	Drehimpuls bezüglich eines Punktes \mathcal{O}
\mathcal{M}	Hilfsmatrix, die während der Initialisierung bei der Steuerstrategie Autoadapt von Grundfos verwendet wird
M^{Bm}	Anzahl der Teilintervalle, mit der die Zeit bei der Konstruktion einer Lösung nach Bellman eingeteilt wird
M^{id}	Anzahl aller Zeitpunkt, zu denen Messwerte für die Identifikation der Parameter das Benchmark-Modell zur Verfügung stehen
M_{res}	Drehmoment der resultierenden Kraft F_{res}
M_{w}	Molare Masse von Wasser
N	Natürliche Zahl als Bezeichnung für eine Anzahl
N	Anzahl aller Testpaare $e^{(i)}$, die bei der Initialisierung der Steuerphase der neuen Steuerstrategie mittels KI benutzt werden
$\mathcal{N}(0, 1)$	Standardnormalverteilung
$N_{\text{tp}}^{\text{AI}}$	Hyperparameter der neuen Steuerstrategie mittels KI, welcher die Anzahl der Epochen für das Training des neuronalen Netzes am Ende der Testphase angibt
N^{Bm}	Anzahl der Teilintervalle, mit der die Drehzahl bei der Konstruktion einer Lösung nach Bellman eingeteilt wird
N_a	Anzahl der Testaktionen, die bei der Initialisierung der Steuerphase der neuen Steuerstrategie mittels KI benutzt werden

N_{msr}	Anzahl aller messbaren Zustandsgrößen des Benchmark-Modells
N_s	Anzahl der Testzustände, die bei der Initialisierung der Steuerphase der neuen Steuerstrategie mittels KI benutzt werden
P_Q	Leistung der Wärmezufuhr
P_{ext}	Leistung aller von außen an einem Körper angreifenden Kräfte
Q	Wärmeenergie
Q	Matrix aller Ausgangsskalare $q^{(i)}$ des Testdatensatzes, der bei der Initialisierung der Steuerphase der neuen Steuerstrategie mittels KI erstellt wird
\tilde{Q}_*	Für die Batch Formulierung benutzte Matrix der Ausgabeskalare des neuronalen Netzes für die Näherung der optimalen Action-Value-Funktion bei der neuen Steuerstrategie mittels KI
\dot{Q}_{add}	Wärmestrom der durch alle zusätzlichen Wärmequellen im Raum verursacht wird
$\dot{Q}_{\text{ceil,ext}}$	Wärmestrom von der Decke nach außen
$\dot{Q}_{\text{ceil,int}}$	Wärmestrom zwischen der Decke und dem Raum
$\dot{Q}_{\text{fc,ext}}$	Zusammengefasster Wärmestrom des Wärmestroms vom Fußboden nach außen und des Wärmestroms von der Decke nach außen
$\dot{Q}_{\text{fc,int}}$	Zusammengefasster Wärmestrom vom Wärmestrom zwischen dem Fußboden und dem Raum und dem Wärmestrom zwischen der Decke und dem Raum
$\dot{Q}_{\text{floor,ext}}$	Wärmestrom vom Fußboden nach außen
$\dot{Q}_{\text{floor,int}}$	Wärmestrom zwischen dem Fußboden und dem Raum
\dot{Q}_{htr}	Wärmeleistung eines Heizkörpers
$\dot{Q}_{\text{htr},i}$	Wärmeleistung des i -ten Segments eines Heizkörpers
$\dot{Q}_{\text{htr,nom}}$	Norm-Wärmeleistung eines Heizkörpers
\dot{Q}_{hw}	Wärmestrom, der sich aufgrund des durch einen Heizkörper fließenden Heizwassers ergibt
$\dot{Q}_{\text{hw},i}$	Wärmestrom, der sich aufgrund des durch das i -te Segment eines Heizkörpers fließenden Heizwassers ergibt
$\dot{Q}_{\text{wall,ext}}$	Wärmestrom von der Mitte einer Wand zu ihrer Außenseite
$\dot{Q}_{\text{wall,int}}$	Wärmestrom von der Mitte einer Wand zu ihrer Innenseite
\dot{Q}_{win}	Wärmestrom durch alle Fenster einer Wand nach außen
R	Hydraulischer Widerstand
\mathcal{R}	Menge aller möglichen Belohnungen, die ein Agent nach Ausführung einer Aktion a erhalten kann
$\bar{R}(\cdot)$	Funktion für die Durchschnittsbelohnung bei der Reinforcement Learning Formulierung

$\hat{R}(\cdot)$	Fiktive Funktion für die Durchschnittsbelohnung, die bei der Initialisierung der Steuerphase der neuen Steuerstrategie mittels KI benutzt wird
$R_\tau^{(1)}$	Erste Teilbelohnung für die Gewichtung der Gesamtbelohnung R_τ , die der Agent bei der Steuerung mittels KI im τ -ten Zeitschritt erhält
$R_\tau^{(2)}$	Zweite Teilbelohnung für die Gewichtung der Gesamtbelohnung R_τ , die der Agent bei der Steuerung mittels KI im τ -ten Zeitschritt erhält
R_{\max}	Parameter der neuen Steuerstrategie mittels KI, der für eine ähnliche Skalierung der Teilbelohnungen $R_\tau^{(1)}$ und $R_\tau^{(2)}$ sorgt
R_τ	Belohnung, die der Agent bei der Steuerung mittels KI im τ -ten Zeitschritt erhält
\hat{R}_τ	Fiktive Belohnung, die bei der Initialisierung der Steuerphase der neuen Steuerstrategie mittels KI benutzt wird
Re	Reynolds-Zahl
Re_1	Untere Grenze der Reynolds-Zahl für den Übergangsbereich von laminaren zu turbulenten Strömungen
Re_2	Obere Grenze der Reynolds-Zahl für den Übergangsbereich von laminaren zu turbulenten Strömungen
Re_{krit}	Kritische Reynolds-Zahl
S	Entropie
\mathcal{S}	Menge aller möglichen Zustände, die das Benchmark-System annehmen kann
S_1	Hilfspunkt auf K_{\max} für die geometrische Interpretation bei der Steuerstrategie Autoadapt von Grundfos
\tilde{S}_1	Hilfspunkt auf K_{\max} für die geometrische Interpretation bei der Steuerstrategie Autoadapt von Grundfos
S_2	Hilfspunkt auf K_{\min} für die geometrische Interpretation bei der Steuerstrategie Autoadapt von Grundfos
\tilde{S}_2	Hilfspunkt auf K_{\min} für die geometrische Interpretation bei der Steuerstrategie Autoadapt von Grundfos
S_τ	Systemzustand, den das Benchmark-Modell bei der Steuerung mittels KI im τ -ten Zeitschritt annimmt
T	Absolute Temperatur
ΔT	Regeldifferenz eines Thermostatventils. Differenz zwischen der Raumtemperatur und der Wunschtemperatur des Nutzers
T_0	Nullpunkt der Celsius-Temperaturskala in Kelvin
T_{air}	Lufttemperatur (Raum- oder Außentemperatur), die die Innen- oder Außenseite einer Wand umgibt
T_{ceil}	Deckentemperatur

T_{fc}	Mittelwert aus der Fußboden- und der Deckentemperatur
$T_{fc,0}$	Mittelwert aus der Fußboden- und der Deckentemperatur zum Zeitpunkt t_0
T_{floor}	Fußbodentemperatur
T_{flw}	Vorlauftemperatur
$T_{flw,0}$	Vorlauftemperatur zum Zeitpunkt t_0
$T_{flw,nom}$	Norm-Vorlauftemperatur nach DIN EN 442
$T_{hc,0}$	Ordinatenabschnitt der als linearen Funktion vereinfachten Heizkurve
T_{htr}	Temperatur des Heizkörpers
$T_{htr,0}$	Temperatur des Heizkörpers zum Zeitpunkt t_0
$T_{htr,i}$	Temperatur des i -ten Segments eines Heizkörpers
ΔT_{ln}	Logarithmische Übertemperatur
T_{room}	Raumtemperatur
$T_{room,0}$	Raumtemperatur zum Zeitpunkt t_0
$T_{room,nom}$	Norm-Bezugsluft-Temperatur des Raumes nach DIN EN 442
T_{rtn}	Rücklauftemperatur
$T_{rtn,0}$	Rücklauftemperatur zum Zeitpunkt t_0
$T_{rtn,nom}$	Norm-Rücklauftemperatur nach DIN EN 442
T_{out}	Außentemperatur
T_{set}	Solltemperatur des Raumes
T_{tol}	Toleranz für die Abweichung der Raumtemperatur von der Solltemperatur
T_{user}	Wunschtemperatur des Nutzers für den Raum
T_{wall}	Temperatur einer Wand auf der Hälfte ihrer Dicke
$T_{wall,0}$	Temperatur einer Wand auf der Hälfte ihrer Dicke zum Zeitpunkt t_0
$T_{wall,ext}$	Oberflächentemperatur auf der Außenseite einer Wand
$T_{wall,int}$	Oberflächentemperatur der Innenseite einer Wand
U	Innere Energie
$\mathcal{U}(a, b)$	Stetige Gleichverteilung auf dem Intervall $[a, b]$
U_{test}	Vereinigung aller Zeitintervalle, in denen für eine Steuerstrategie Drehzahltests durchgeführt werden
U_{win}	Wärmedurchgangskoeffizient aller Fenster einer Wand
V	Volumen eines Körpers
\dot{V}	Volumenstrom
$\Delta \dot{V}^{Gf}$	Parameter für die Steuerstrategie Autoadapt von Grundfos, der bei der Neuberechnung der Volumenströme $\dot{V}_{G_{min}}$ und $\dot{V}_{G_{max}}$ benutzt wird
$\Delta \dot{V}^{KSB}$	Toleranz für den Volumenstrom bei der Steuerstrategie Dynamic Control von KSB
V_0	Volumen der Ausgangskonfiguration eines Körpers

\dot{V}_0	Volumenstrom zum Zeitpunkt t_0
$\dot{V}_{G_{\max}}$	Volumenstrom des oberen Grenzpunktes G_{\max} auf der Regelkurve K_{mid} bei der Steuerstrategie Autoadapt von Grundfos
$\dot{V}_{G_{\min}}$	Volumenstrom des unteren Grenzpunktes G_{\min} auf der Regelkurve K_{mid} bei der Steuerstrategie Autoadapt von Grundfos
\dot{V}_{S_1}	Volumenstrom des Hilfspunktes S_1 für die geometrische Interpretation bei der Steuerstrategie Autoadapt von Grundfos
\dot{V}_{S_2}	Volumenstrom des Hilfspunktes S_2 für die geometrische Interpretation bei der Steuerstrategie Autoadapt von Grundfos
\dot{V}_e	Volumenstrom am Endpunkt eines Rohres
$\dot{V}_{a_{\max}}$	Volumenstrom des Schnittpunkts zwischen der maximalen Pumpenkennlinie und der maximalen Anlagenkennlinie bei der neuen Steuerstrategie mittels KI
$\dot{V}_{a_{\min}}$	Volumenstrom des Schnittpunkts zwischen der maximalen Pumpenkennlinie und der minimalen Anlagenkennlinie bei der neuen Steuerstrategie mittels KI
$\dot{V}_{a_{\text{sys}}}$	Volumenstrom des Schnittpunkts zwischen der maximalen Pumpenkennlinie und der aktuellen Anlagenkennlinie bei der neuen Steuerstrategie mittels KI
\dot{V}_{dsgn}	Volumenstrom des Auslegungspunktes, welcher sich bei einem idealen hydraulischen Abgleich des Heizsystems ergeben würde
\dot{V}_{ref}	Referenzvolumenstrom
\dot{V}_{rs}	Volumenstrom zum Zeitpunkt des letzten Resets bei der Steuerstrategie Dynamic Control von KSB
\dot{V}_s	Volumenstrom am Anfangspunkt eines Rohres
W	Gesamtenergie
$W^{[l]}$	Matrix der Gewichte der Verbindungen zwischen der $l - 1$ -ten und l -ten Schicht des neuronalen Netzes für die Näherung der optimalen Action-Value-Funktion bei der neuen Steuerstrategie mittels KI
\mathcal{X}	Die aktuelle Konfiguration aller materiellen Punkte eines Körpers
X_A	Zufallsvariable, die für die Definition von A_τ bei der neuen Steuerstrategie mittels KI benutzt wird
X_ϵ	Zufallsvariable, die für die Definition von A_τ bei der neuen Steuerstrategie mittels KI benutzt wird
Y	Allgemeine mechanische Feldgröße
$Z^{[l]}$	Für die Batch Formulierung benutzte Matrix der Voraktivierungen der l -ten Schicht des neuronalen Netzes für die Näherung der optimalen Action-Value-Funktion bei der neuen Steuerstrategie mittels KI
a	Variable für eine beliebige Aktion bei der Reinforcement Learning Formulierung

\mathbf{a}	Positionsvektor eines materiellen Teilchens
$a_{\text{tp}}^{(k,j)}$	Testaktionen, die während der Haupttestphase der neuen Steuerstrategie mittels KI durchgeführt werden
a^{QP}	Stauchungs- bzw. Streckungsfaktor einer quadratischen Regelkurve
a_1	Hilfsgröße für die Beschreibung der Bestimmungsgleichung von a_{mid} bei der Steuerstrategie Autoadapt von Grundfos
a_2	Hilfsgröße für die Beschreibung der Bestimmungsgleichung von a_{mid} bei der Steuerstrategie Autoadapt von Grundfos
a_3	Hilfsgröße für die Beschreibung der Bestimmungsgleichung von a_{mid} bei der Steuerstrategie Autoadapt von Grundfos
a_{max}	Stauchungs- bzw. Streckungsfaktor der maximalen Anlagenkennlinie K_{max} bei der Steuerstrategie Autoadapt von Grundfos
a_{max}	Stauchungs- bzw. Streckungsfaktor der maximalen Anlagenkennlinie
\tilde{a}_{max}	Näherung für den Stauchungs- bzw. Streckungsfaktor der maximalen Anlagenkennlinie bei der neuen Steuerstrategie mittels KI
a_{mid}	Stauchungs- bzw. Streckungsfaktor der quadratischen Regelkurve K_{mid} bei der Steuerstrategie Autoadapt von Grundfos
a_{min}	Stauchungs- bzw. Streckungsfaktor der minimalen Anlagenkennlinie K_{min} bei der Steuerstrategie Autoadapt von Grundfos
a_{min}	Stauchungs- bzw. Streckungsfaktor der minimalen Anlagenkennlinie
\tilde{a}_{min}	Näherung für den Stauchungs- bzw. Streckungsfaktor der minimalen Anlagenkennlinie bei der neuen Steuerstrategie mittels KI
a_{sys}	Stauchungs- bzw. Streckungsfaktor der parabelförmigen Anlagenkennlinie
\mathbf{b}	Hilfsvektor, der während der Initialisierung bei der Steuerstrategie Autoadapt von Grundfos verwendet wird
$\mathbf{b}^{[l]}$	Vektor der additiven Bias zwischen der $l - 1$ -ten und l -ten Schicht des neuronalen Netzes für die Näherung der optimalen Action-Value-Funktion bei der neuen Steuerstrategie mittels KI
b_1	Hilfsgröße für die Beschreibung der Bestimmungsgleichung von a_{mid} bei der Steuerstrategie Autoadapt von Grundfos
b_2	Hilfsgröße für die Beschreibung der Bestimmungsgleichung von a_{mid} bei der Steuerstrategie Autoadapt von Grundfos
b_3	Hilfsgröße für die Beschreibung der Bestimmungsgleichung von a_{mid} bei der Steuerstrategie Autoadapt von Grundfos

c_V	Spezifische Wärmekapazität von Wasser bei konstantem Volumen
c_p	Spezifische Wärmekapazität von Wasser bei konstantem Druck
$c_{vc,0}$	Parameter für die Beschreibung des nichtlinearen Teils der vereinfachten Ventilkennlinie
$c_{vc,1}$	Parameter für die Beschreibung des nichtlinearen Teils der vereinfachten Ventilkennlinie
$c_{vc,2}$	Parameter für die Beschreibung des nichtlinearen Teils der vereinfachten Ventilkennlinie
$c_{vc,3}$	Parameter für die Beschreibung des nichtlinearen Teils der vereinfachten Ventilkennlinie
c_w	Spezifische Wärmekapazität von Wasser
\mathbf{d}	Vektor aller Störgrößen des Benchmark-Modells
d_1	Hilfsstrecke zwischen S_1 und G_{\max} für die geometrische Interpretation bei der Steuerstrategie Autoadapt von Grundfos
d_2	Hilfsstrecke zwischen S_2 und G_{\max} für die geometrische Interpretation bei der Steuerstrategie Autoadapt von Grundfos
d_{pipe}	Innendurchmesser eines Rohres
d_{wall}	Dicke einer Wand
\mathbf{e}	Eingangsvektor des neuronalen Netzes für die Näherung der optimalen Action-Value-Funktion bei der neuen Steuerstrategie mittels KI
$\mathbf{e}^{(i)}$	Eingangsvektoren für den Testdatensatz, der bei der Initialisierung der Steuerphase der neuen Steuerstrategie mittels KI erstellt wird
$\mathbf{e}_{\text{tp}}^{(k,j)}$	Während der Haupttestphase der neuen Steuerstrategie mittels KI gesammelte Daten, die für das Weitertrainieren des neuronalen Netzes als Eingangsvektoren benutzt werden
\mathbf{f}	Vektorwertige Funktion zur Beschreibung aller differentiellen Gleichungen des Benchmark-Modells
f_{rad}	Der durch die Wärmestrahlung verursachte Anteil der Leistungsabgabe eines Heizkörpers
g	Schwerebeschleunigung
\mathbf{g}	Vektorwertige Funktion zur Beschreibung aller algebraischen Gleichungen des Benchmark-Modells
g_{hc}	Funktion zur Beschreibung der Heizkurve
g_{opp}	Funktion zur Bestimmung des Volumenstroms des aktuellen Betriebspunktes einer Pumpe
g_{vc}	Funktion zur Beschreibung der Ventilkennlinie
h	Höhenlage eines Rohrpunktes
Δh	Höhendifferenz des Anfangs- und Endpunktes eines Rohres
\mathbf{h}	Wärmestromvektor pro Einheitsfläche

$\mathbf{h}^{[l]}$	Vektor der Aktivierungen der l -ten Schicht des neuronalen Netzes für die Näherung der optimalen Action-Value-Funktion bei der neuen Steuerstrategie mittels KI
h_e	Höhenlage des Endpunktes eines Rohres
h_{ext}	Wärmeübergangskoeffizient einer Wandaußenseite
h_{int}	Wärmeübergangskoeffizient einer Wandinnenseite
h_s	Höhenlage des Anfangspunktes eines Rohres
h_{wall}	Höhe einer Wand
i	Natürliche Zahl als Bezeichnung für einen Index
j	Natürliche Zahl als Bezeichnung für einen Index
k	Natürliche Zahl als Bezeichnung für einen Index
\mathbf{k}	Massenkraftdichte
k_{pipe}	Absolute Wandrauigkeit
k_{vs}	Durchflusskoeffizient des maximal geöffneten Ventils
l	Natürliche Zahl als Bezeichnung für einen Index
l_0	Parameter der neuen Steuerstrategie mittels KI für die Berechnung von $\tilde{\alpha}(\cdot)$
l_1	Parameter der neuen Steuerstrategie mittels KI für die Berechnung von $\tilde{\alpha}(\cdot)$
l_{pipe}	Länge eines Rohres
l_{wall}	Länge einer Wand
m	Masse eines Körpers
m^{PP}	Steigung einer linearen Regelkurve
m_{hc}	Steigung der als linearen Funktion vereinfachten Heizkurve
m_{pc}	Hilfsgröße bei der Berechnung von g_{opp}
m_{vc}	Steigung des linearen Teils der vereinfachten Ventilkennlinie
n	Drehzahl einer Pumpe
\mathbf{n}	Einheitsnormalenvektor
$n_{\text{test}}^{(k,j)}$	Testdrehzahl, die während der Haupttestphase der neuen Steuerstrategie mittels KI angesteuert wird
$\Delta n_{\text{max}}^{\text{AI}}$	Betragsmäßig größte positive Drehzahländerung, die während eines Steuerschritts bei der neuen Steuerstrategie mittels KI vorgenommen werden kann
$\Delta n_{\text{min}}^{\text{AI}}$	Betragsmäßig größte negative Drehzahländerung, die während eines Steuerschritts bei der neuen Steuerstrategie mittels KI vorgenommen werden kann
Δn^{Bm}	Drehzahlschrittweite, die bei der Konstruktion einer Lösung nach Bellman benutzt wird
n^{KSB}	Parameter, mit dessen Hilfe die dynamische Drehzahlabenkung bei der Steuerstrategie Dynamic Control von KSB beschrieben wird

$\tilde{n}^{[0]}$	Anzahl der Neuronen der Eingangsschicht des neuronalen Netzes, das bei der Initialisierung der Steuerphase der neuen Steuerstrategie mittels KI trainiert wird
$\tilde{n}^{[L]}$	Anzahl der Neuronen der letzten Schicht des neuronalen Netzes, das bei der Initialisierung der Steuerphase der neuen Steuerstrategie mittels KI trainiert wird
$n^{[l]}$	Anzahl der Neuronen der l -ten Schicht des neuronalen Netzes für die Näherung der optimalen Action-Value-Funktion bei der neuen Steuerstrategie mittels KI
n^{cs}	Feste Drehzahl bei einer Regelung mit konstanter Drehzahl
n_{htr}	Heizkörperexponent
n_{max}	Größte Drehzahl, die von einer Pumpe angesteuert werden kann
n_{min}	Kleinste Drehzahl, die von einer Pumpe angesteuert werden kann
n_{pump}	Drehzahl der Pumpe
$n_{\text{pump},0}$	Drehzahl der Pumpe zum Zeitpunkt t_0
n_{ref}	Referenzdrehzahl
n_{u}	Eine zulässige, beliebige Drehzahl, die von einer Pumpe angesteuert werden kann
o_{dsgn}	Parameter, der einen Sicherheitsaufschlag für die Parameter der elementaren Steuerstrategien widerspiegelt
p	Druck, der an einem Rohrpunkt herrscht
Δp	Druckdifferenz
\mathbf{p}	Impulsvektor
$p(\cdot, \cdot \mid \cdot, \cdot)$	Funktion für die Beschreibung der Dynamik bei der Reinforcement Learning Formulierung
$p_{\text{air,nom}}$	Norm-Luftdruck
p_{e}	Druck, der am Endpunkt eines Rohres herrscht
Δp_{f}	Durch die Reibung beim Durchströmen eines Rohres verursachter Druckverlust
Δp_{pipe}	Druckverlust, der beim Durchströmen eines Rohres entsteht
$\Delta p_{\text{pipe,l}}$	Druckverlust, der beim Durchströmen eines Rohres entsteht, wenn die Strömung laminar ist
$\Delta p_{\text{pipe,t}}$	Druckverlust, der beim Durchströmen eines Rohres entsteht, wenn die Strömung turbulent ist
Δp_{ref}	Referenzwert des Druckverlustes für die Definition von k_{vs}
p_{s}	Druck, der am Anfangspunkt eines Rohres herrscht
$q^{(i)}$	Ausgangsskalare für den Testdatensatz, der bei der Initialisierung der Steuerphase der neuen Steuerstrategie mittels KI erstellt wird
$\tilde{q}_*^{(i)}$	Ausgangsskalare des neuronalen Netzes bei Benutzung von $e^{(i)}$ als Eingangsvektoren, die bei Initialisierung der Steuerphase der neuen Steuerstrategie mittels KI berechnet werden

$q_{\text{tp}}^{(k,j)}$	Während der Haupttestphase der neuen Steuerstrategie mittels KI gesammelte Daten, die für das Weitertrainieren des neuronalen Netzes als Ausgangsskalare benutzt werden
q^{Gf}	Parameter der Steuerstrategie Autoadapt von Grundfos, der bei der Berechnung von $\dot{V}_{G_{\min}}$ und $\dot{V}_{G_{\max}}$ verwendet wird
$q_*(\cdot, \cdot)$	Optimale Action-Value-Funktion bei der Reinforcement Learning Formulierung
$\hat{q}_*(\cdot, \cdot)$	Fiktive optimale Action-Value-Funktion, die bei der Initialisierung der Steuerphase der neuen Steuerstrategie mittels KI benutzt wird
$\tilde{q}_*(\cdot, \cdot)$	Durch ein neuronales Netz parametrisierte Näherung für die optimale Action-Value-Funktion der neuen Steuerstrategie mittels KI
$q_\pi(\cdot, \cdot)$	Action-Value-Funktion bezüglich der Policy π bei der Reinforcement Learning Formulierung
r	Spezifische Dichte der inneren Wärmequellen
r	Verhältnis zwischen einer zulässigen, beliebigen Drehzahl und der Referenzdrehzahl
r	Variable für eine Belohnung bei der Reinforcement Learning Formulierung
$t_{\text{tp}}^{(k,j)}$	Testbelohnungen, die sich während der Haupttestphase der neuen Steuerstrategie mittels KI ergeben
$\bar{r}_{\text{tp}}^{(k,j)}$	Durchschnittsbelohnungen, die sich während der Haupttestphase der neuen Steuerstrategie mittels KI ergeben
s	Spezifische innere Entropie
s	Am Thermostatkopf vom Nutzer eingestellter Skalenwert
s	Variable für einen beliebigen Zustand des Systems bei der Reinforcement Learning Formulierung
s'	Variable für einen beliebigen Zustand des Systems bei der Reinforcement Learning Formulierung
$s_{\text{tp}}^{(k,j)}$	Testzustände, die sich während der Haupttestphase der neuen Steuerstrategie mittels KI ergeben
t	Zeit
$t_{\text{test}}^{(k,j)}$	Zeitpunkt, an dem während der Haupttestphase der neuen Steuerstrategie mittels KI eine Testdrehzahl angenommen wird
Δt^{AI}	Zeitlicher Abstand zwischen zwei Steuerzeitpunkten t_τ^{AI} und $t_{\tau+1}^{\text{AI}}$ bei der eigentlichen Steuerphase der neuen Steuerstrategie mittels KI
t_0^{AI}	Zeitpunkt, an dem die eigentliche Steuerphase der neuen Steuerstrategie mittels KI beginnt
$\Delta t_{\text{mxmn}}^{\text{AI}}$	Länge der $n_{\max} - n_{\min}$ -Phase der neuen Steuerstrategie mittels KI

Δt_{tp}^{AI}	Länge eines Drehzahltests während der Haupttestphase der neuen Steuerstrategie mittels KI
Δt_{trans}^{AI}	Zeitspanne, in der während der Testphase der neuen Steuerstrategie mittels KI ein stetiger Übergang zwischen zwei Drehzahlen stattfindet
t_{τ}^{AI}	Steuerzeitpunkt der eigentlichen Steuerphase der neuen Steuerstrategie mittels KI
Δt^{Bm}	Zeitspanne, die bei der Konstruktion einer Lösung nach Bellman benutzt wird
t_j^{Bm}	Äquidistante Stützpunkte für die Zeitintervalle, die bei der Konstruktion einer Lösung des Steuerproblems nach dem Optimalitätsprinzip von Bellman benutzt werden
Δt^{Gf}	Zeitspanne, die während der Initialisierung und bei einem Drehzahltest bei der Steuerstrategie Autoadapt von Grundfos verwendet wird
Δt^{KSB}	Zeitspanne, mit dessen Hilfe der Beginn der Drehzahlabenkung bei der Steuerstrategie Dynamic Control von KSB charakterisiert wird
t_f^{id}	Endzeit für die Systemidentifikation
t_j^{id}	Zeitpunkte, zu denen Messwerte für die messbaren Zustandsgrößen des Benchmark-Modells zur Verfügung stehen
Δt_{μ}^{id}	Abstand zwischen zwei Zeitpunkten, zu denen „Messwerte“ für das Mehrfachschießverfahren vorliegen
t_0	Anfangszeit
t_1	Hilfszeitpunkt für die Initialisierung bei der Steuerstrategie Autoadapt von Grundfos
t_2	Hilfszeitpunkt für die Initialisierung bei der Steuerstrategie Autoadapt von Grundfos
t_3	Hilfszeitpunkt für die Initialisierung bei der Steuerstrategie Autoadapt von Grundfos
t_4	Hilfszeitpunkt für die Initialisierung bei der Steuerstrategie Autoadapt von Grundfos
t_5	Hilfszeitpunkt für die Initialisierung bei der Steuerstrategie Autoadapt von Grundfos
t_6	Hilfszeitpunkt für die Initialisierung bei der Steuerstrategie Autoadapt von Grundfos
t_f	Endzeit
t_{im}	Beliebiger Zeitpunkt, der zwischen t_0 und t_f liegt
t_{low}	Zeitspanne, seit der die Absenkung bei der Steuerstrategie Dynamic Control von KSB aktiv ist
t_{rs}	Zeitpunkt des letzten Resets bei der Steuerstrategie Dynamic Control von KSB

$t_{rs,new}$	Neuer Zeitpunkt des letzten Resets bei der Steuerstrategie Dynamic Control von KSB
t_{test}	Zeitpunkt des Beginns eines Drehzahltestes bei der Steuerstrategie Autoadapt von Grundfos
u	Spezifische innere Energie
u	Steuergröße des Benchmark-Modells
\tilde{u}_*^{Bm}	Mithilfe des Optimalitätsprinzips von Bellman ermittelte Näherung für die Lösung des optimalen Steuerproblems
$u^{Bm,(j)}$	Optimale Lösung des j -ten Teilproblems für das Steuerproblem, welches sich aus dem Optimalitätsprinzip von Bellman ergibt
$\tilde{u}^{Bm,(j)}$	Näherung für die Lösung des j -ten Teilproblems für das Steuerproblem, welches sich aus dem Optimalitätsprinzip von Bellman ergibt
u_{dyn}^{KSB}	Dynamischer Anteil der Steuerfunktion bei der Steuerstrategie Dynamic Control von KSB
u_{stat}^{KSB}	Statischer Anteil der Steuerfunktion bei der Steuerstrategie Dynamic Control von KSB
u^{CP}	Steuerfunktion für eine Pumpe, die mit einer Konstantdruckregelung betrieben wird
u^{CS}	Steuerfunktion für eine ungeregelte Pumpe, die mit einer konstanten Drehzahl betrieben wird
u^{PP}	Steuerfunktion für eine Pumpe, die mit einer Proportionaldruckregelung betrieben wird
u^{QP}	Steuerfunktion für eine Pumpe, die mit einer Regelung auf einer quadratischen Regelkurve betrieben wird
u_*	Lösung des optimalen Steuerproblems
u_0	Anfangswert für die Steuergröße des Benchmark-Modells
u_j	Parameter für die Definition der Steuerfunktionen $\tilde{u}^{Bm,(j)}$, die bei der Konstruktion einer Lösung nach Bellman benutzt werden
v	Mittlere Strömungsgeschwindigkeit
\mathbf{v}	Geschwindigkeitsvektor
$v_*(\cdot)$	Optimale Value-Funktion bei der Reinforcement Learning Formulierung
v_e	Strömungsgeschwindigkeit am Endpunkt eines Rohres
v_s	Strömungsgeschwindigkeit am Anfangspunkt eines Rohres
$v_\pi(\cdot)$	Value-Funktion bezüglich der Policy π bei der Reinforcement Learning Formulierung
\mathbf{x}	Ortsvektor
\mathbf{x}	Hilfsvektor, die während der Initialisierung bei der Steuerstrategie Autoadapt von Grundfos verwendet wird
\mathbf{x}	Vektor aller Zustandsgrößen des Benchmark-Modells
\mathbf{x}_0	Vektor aller Zustandsgrößen des Benchmark-Modells zum Zeitpunkt t_0

x_r	Lösungsvektor mit allen Zustandsgrößen, der sich aus den K^{id} Lösungen der relaxierten DAE Anfangswertprobleme zusammensetzt
y	Vektor aller differentiellen Zustandsgrößen des Benchmark-Modells
y_0	Vektor aller differentiellen Zustandsgrößen des Benchmark-Modells zum Zeitpunkt t_0
y_r	Lösungsvektor mit allen differentiellen Zustandsgrößen, der sich aus den K^{id} Lösungen der relaxierten DAE Anfangswertprobleme zusammensetzt
z	Vektor aller algebraischen Zustandsgrößen des Benchmark-Modells
$z^{[l]}$	Vektor der Voraktivierungen der l -ten Schicht des neuronalen Netzes für die Näherung der optimalen Action-Value-Funktion bei der neuen Steuerstrategie mittels KI
z_0	Vektor aller algebraischen Zustandsgrößen des Benchmark-Modells zum Zeitpunkt t_0
z_r	Lösungsvektor mit allen algebraischen Zustandsgrößen, der sich aus den K^{id} Lösungen der relaxierten DAE Anfangswertprobleme zusammensetzt
Φ	Oberflächendichte der äußeren Einwirkungen auf einen Körper
$\tilde{\Phi}$	Oberflächendichte der äußeren Einwirkungen auf einen Körper
Ξ	Massendichte der äußeren Einwirkungen auf einen Körper
$\tilde{\Xi}$	Volumendichte der äußeren Einwirkungen auf einen Körper
Ψ	Massendichteverteilung einer allgemeinen mechanischen Feldgröße
$\tilde{\Psi}$	Volumendichteverteilung einer allgemeinen mechanischen Feldgröße
α	Ventilöffnungsgrad
$\tilde{\alpha}(\cdot)$	Näherung für den Ventilöffnungsgrad $\alpha(\cdot)$ bei der neuen Steuerstrategie mittels KI
$\alpha_{\text{ctrl}}^{\text{AI}}$	Hyperparameter der neuen Steuerstrategie mittels KI, welcher als Lernrate für die Neuberechnungen von $b^{[l]}$ und $W^{[l]}$ in der eigentlichen Steuerphase, dient
$\alpha_{\text{tp}}^{\text{AI}}$	Hyperparameter der neuen Steuerstrategie mittels KI, welcher als Lernrate für die Werte $q_{\text{tp}}^{(k,j)}$, die in der Haupttestphase ermittelt werden, dient
α_0	Ventilöffnungsgrad zum Zeitpunkt t_0
α_1	Parameter der neuen Steuerstrategie mittels KI, der von der Ventilkennlinie abhängt und bei der Definition von $R_\tau^{(2)}$ benutzt wird

α_2	Parameter der neuen Steuerstrategie mittels KI, der von der Ventilkennlinie abhängt und bei der Definition von $R_\tau^{(2)}$ benutzt wird
α_3	Parameter der neuen Steuerstrategie mittels KI, der von der Ventilkennlinie abhängt und bei der Definition von $R_\tau^{(2)}$ benutzt wird
α_4	Parameter der neuen Steuerstrategie mittels KI, der von der Ventilkennlinie abhängt und bei der Definition von $R_\tau^{(2)}$ benutzt wird
α_{dsgn}	Ventilöffnungsgrad, welcher sich bei einer verschwindenden Regeldifferenz des Thermostatventils einstellen würde
α_{min}	Minimaler Ventilöffnungsgrad
$\alpha_{\text{vc},0}$	Ordinatenabschnitt des linearen Teils der vereinfachten Ventilkennlinie
$\beta^{(k)}$	Vektor der Anfangswerte des Mehrfachschießverfahrens auf dem $k + 1$ -ten Intervall
$\beta_{\text{ctrl}}^{\text{AI}}$	Hyperparameter der neuen Steuerstrategie mittels KI, welcher als Lernrate für die Durchschnittsbelohnung \bar{R} in der eigentlichen Steuerphase, dient
$\beta_{\text{tp}}^{\text{AI}}$	Hyperparameter der neuen Steuerstrategie mittels KI, welcher als Lernrate für die Durchschnittsbelohnungen $r_{\text{tp}}^{(k,j)}$, die in der Haupttestphase ermittelt werden, dient
β_R	Gewichtsfaktor, der bei der Gewichtung von $R_\tau^{(1)}$ und $R_\tau^{(2)}$ für die Definition von R_τ eingesetzt wird
γ	Diskontierungsfaktor, der bei der Definition des zukünftigen Gewinns G_τ durch eine gewichtete Summe aller noch ausstehenden Belohnungen eingesetzt wird
δ	Hilfsgrößen, die während der eigentlichen Steuerphase der neuen Steuerstrategie mittels KI für die Berechnung von \bar{R} sowie $b^{[l]}$ und $\mathbf{W}^{[l]}$ benutzt wird
$\delta^{(k)}$	Funktion zur Verbesserung der Konsistenz der algebraischen Bedingungen des Mehrfachschießverfahrens auf dem $k + 1$ -ten Intervall
$\delta_{\text{tp}}^{(k,j)}$	Hilfsgrößen, die während der Haupttestphase der neuen Steuerstrategie mittels KI für die Berechnung von $\bar{r}_{\text{tp}}^{(k,j)}$ und $q_{\text{tp}}^{(k,j)}$ benutzt werden
$\varepsilon_{\text{ctrl}}^{\text{AI}}$	Hyperparameter der neuen Steuerstrategie mittels KI, der bei der Definition von A_τ in der eigentlichen Steuerphase die Wahrscheinlichkeit angibt, dass eine zufällige Aktion ausgewählt wird
η	Dynamischen Viskosität
η_w	Viskosität von Wasser

θ	Vektor aller Parameter des Benchmark-Modells
θ^{id}	Vektor aller Parameter des Benchmark-Modells, die für das Optimierungsproblem zur System-Identifikation als unbekannt gelten.
$\theta_{\text{res}}^{\text{id}}$	Beste gefundene Näherung für die Lösung des Optimierungsproblems der System-Identifikation.
θ_{a}	Vektor aller Parameter des Benchmark-Modells, die in den differentiellen Gleichungen vorkommen
θ_{d}	Vektor aller Parameter des Benchmark-Modells, die in den differentiellen Gleichungen vorkommen
θ_{hc}	Vektor aller Parameter des Benchmark-Modells, die in der Definition der Funktion g_{hc} vorkommen
θ_{opp}	Vektor aller Parameter des Benchmark-Modells, die in der Definition der Funktion g_{opp} vorkommen
θ_{vc}	Vektor aller Parameter des Benchmark-Modells, die in der Definition der Funktion g_{vc} vorkommen
κ	Natürliche Zahl als Bezeichnung für einen Zeitindex bei der Reinforcement Learning Formulierung
$\lambda(\cdot)$	Lebesgue-Maß
λ_{ext}	Wärmeleitfähigkeit einer Wandaußenseite
λ_{f}	Rohrreibungszahl
λ_{int}	Wärmeleitfähigkeit einer Wandinnenseite
λ_{w}	Wärmeleitfähigkeit von Wasser
μ	Matrix aller Messwerte der messbaren Zustandsgrößen des Benchmark-Modells
$\nu_{\text{ceil,ext}}$	Wärmeleitwert der Außenseite der Decke
$\nu_{\text{ceil,int}}$	Wärmeleitwert der Innenseite der Decke
$\nu_{\text{fc,ext}}$	Mittelwert aus den Wärmeleitwerten der Außenseiten des Fußbodens und der Decke
$\nu_{\text{fc,int}}$	Mittelwert aus den Wärmeleitwerten der Innenseiten des Fußbodens und der Decke
$\nu_{\text{floor,ext}}$	Wärmeleitwert der Außenseite des Fußbodens
$\nu_{\text{floor,int}}$	Wärmeleitwert der Innenseite des Fußbodens
ν_{htr}	Wärmeleitwert eines Segments eines Heizkörpers
ν_{w}	Produkt aus der spezifischen Wärmekapazität c_{w} und der Dichte ρ_{w} von Wasser
$\nu_{\text{wall,ext}}$	Wärmeleitwert einer Wandaußenseite
$\nu_{\text{wall,int}}$	Wärmeleitwert einer Wandinnenseite
ν_{win}	Wärmeleitwert aller Fenster einer Wand
$\pi(\cdot)$	Deterministische Policy bei der Reinforcement Learning Formulierung
$\tilde{\pi}(\cdot)$	Deterministische Policy bei der Reinforcement Learning Formulierung

$\pi(\cdot \cdot)$	Stochastische Policy bei der Reinforcement Learning Formulierung
$\pi_*(\cdot)$	Optimale deterministische Policy bei der Reinforcement Learning Formulierung
$\hat{\pi}_*(\cdot)$	Fiktive optimale deterministische Policy, die bei der Initialisierung der Steuerphase der neuen Steuerstrategie mittels KI benutzt wird
ρ	Massendichte eines Körpers
ρ'	Abweichung der Dichte vom Referenzwert ρ_0 eines inkompressiblen Fluids bei Anwendung der Näherung von Boussinesq
ρ_0	Referenzwert der Dichte eines inkompressiblen Fluids bei Anwendung der Näherung von Boussinesq
ρ_{ref}	Referenzwert der Dichte für die Definition von k_{vs}
ρ_{w}	Massendichte von Wasser
σ	Cauchyscher Spannungstensor
$\sigma^{[l]}$	Vektorwertige Aktivierungsfunktion für die Ausgabeneuronen l -ten Schicht des neuronalen Netzes für die Näherung der optimalen Action-Value-Funktion bei der neuen Steuerstrategie mittels KI
σ_H	Skalierungsfaktor in H -Richtung für die geometrische Interpretation bei der Steuerstrategie Autoadapt von Grundfos
$\sigma_{\dot{V}}$	Skalierungsfaktor in \dot{V} -Richtung für die geometrische Interpretation bei der Steuerstrategie Autoadapt von Grundfos
σ_{ij}	Inverser Gewichtungsfaktor in der Zielfunktion J^{id} des Mehrfachschießverfahrens für die i -te Zustandsgröße und den j -ten Messzeitpunkt
τ	Natürliche Zahl als Bezeichnung für einen Zeitindex bei der Reinforcement Learning Formulierung
$\Delta\tau^{\text{id}}$	Längen der Intervalle, die für das Mehrfachschießverfahren festgelegt werden
τ_k^{id}	Zeitpunkte, mit denen die Intervalle für das Mehrfachschießverfahren festgelegt werden

Literaturverzeichnis

- [1] AFAQ, Saahil und Smitha RAO. Significance Of Epochs On Training A Neural Network. *International Journal of Scientific & Technology Research*, Bd. 9 (2020), Nr. 6, S. 485–488. IJSTR Publications. ISSN 2277-8616. Verfügbar unter: <https://www.ijstr.org/final-print/jun2020/Significance-Of-Epochs-On-Training-A-Neural-Network.pdf>. [Zugriff am 15. Juli 2025].
- [2] AGGARWAL, Charu C. *Neural Networks and Deep Learning: A Textbook*. E-Book. First Edition. Cham [Schweiz]: Springer, 2018. ISBN 978-3-319-94463-0 [online]. ISBN 978-3-319-94462-3 [Print]. Verfügbar unter: <https://doi.org/10.1007/978-3-319-94463-0>.
- [3] AKIMA, Hiroshi. A New Method of Interpolation and Smooth Curve Fitting Based on Local Procedures. *Journal of the ACM*, Bd. 17 (1970), Nr. 4, S. 589–602. Association for Computing Machinery. Verfügbar unter: <https://doi.org/10.1145/321607.321609>. [Zugriff am 15. Juli 2025]. Auch verfügbar unter: <http://www.leg.ufpr.br/lib/exe/fetch.php/wiki:internas:biblioteca:akima.pdf>.
- [4] ALBERS, Karl-Josef (Herausgeber). *Taschenbuch für Heizung und Kilmatechnik: einschließlich Trinkwasser- und Kätetechnik sowie Energiekonzepte* [2 Bde.]. E-Book. 78. Auflage. Hermann RECKNAGEL (Begründer), Eberhard SPRENGER (Fortführer). München: DIV Deutscher Industrieverlag GmbH, 2017. ISBN 978-3-8356-7285-7 [online]. ISBN 978-3-8356-7284-0 [Print]. Verfügbar unter: <http://scifo.de/9783835672857>.
- [5] ALTENBACH, Holm. *Kontinuumsmechanik: Einführung in die materialunabhängigen und materialabhängigen Gleichungen*. E-Book. 4., korrigierte und überarbeitete Auflage. Berlin: Springer Vieweg, 2018. ISBN 978-3-662-57504-8 [online]. ISBN 978-3-662-57503-1 [Print]. Verfügbar unter: <https://doi.org/10.1007/978-3-662-57504-8>.
- [6] BABUSCH, André; Thomas EBERT; Karl-Heinz KÖNIG; Thomas MAKOSCHEY; Andreas MILLIES et al. (Redaktion). 500/1601/DE, *Grundlagen der Pumpentechnik. [Pumpenfibel]*. PDF. 6. überarbeitete und aktualisierte Auflage. Dortmund: WILO SE, 2016. Verfügbar unter: <https://wilo.cdn.mediamid.com/cdnoc/wilo162172/1004632/wilo162172.pdf>. [Zugriff am 15. Juli 2025].
- [7] BARTLE, Robert G. *The Elements of Integration and Lebesgue Measure*. E-Book. Wiley Classics Library Edition. New York: John Wiley & Sons, Inc., 1995. ISBN 978-1-118-16447-1 [online]. ISBN 978-0-471-04222-8 [Print]. Verfügbar unter: <https://doi.org/10.1002/9781118164471>.

- [8] BATCHELOR, G. K. *An Introduction to Fluid Dynamics*. E-Book. First Cambridge Mathematical Library Edition. Cambridge: Cambridge University Press, 2000. ISBN 978-0-511-80095-5 [online]. ISBN 978-0-521-66396-0 [Print]. Verfügbar unter: <https://doi.org/10.1017/CB09780511800955>.
- [9] BATTISTEL, Laura; Andrea VILARDI; Massimiliano ZAMPINI und Riccardo PARIN. An investigation on humans' sensitivity to environmental temperature. *Scientific Reports*, Bd. 13 (2023), Nr. 1, Article number: 21353. Nature Research. ISSN 2045-2322. Verfügbar unter: <https://doi.org/10.1038/s41598-023-47880-5>. [Zugriff am 15. Juli 2025].
- [10] BATTISTI, Kurt (Herausgeber). *Kalkzementputz (1600)*. Online. In: Basiskatalog Österreich [AT]. 27. Sep. 2008. Verfügbar unter: <http://www.bphdb.com/material/show/1695>. [Zugriff am 15. Juli 2025].
- [11] BATTISTI, Kurt (Herausgeber). *Kalkzementputz, außen (1800)*. Online. In: Basiskatalog Österreich [AT]. 5. Okt. 2008. Verfügbar unter: <http://www.bphdb.com/material/show/1696>. [Zugriff am 15. Juli 2025].
- [12] BATTISTI, Kurt (Herausgeber). *Kalkzementputz, innen (1800)*. Online. In: Basiskatalog Österreich [AT]. 5. Okt. 2008. Verfügbar unter: <http://www.bphdb.com/material/show/1706>. [Zugriff am 15. Juli 2025].
- [13] BAYER, Manfred. Autonome Auswahl der Regelkennlinie: AutoAdapt-System der 2. Generation befreit Installateure von Einstellarbeiten an Heizungsumwälzpumpen. Online. *IKZ-Haustechnik: Magazin für Gebäude- und Energietechnik*, 15. Aug. 2008, Nr. 16, S. 28–30. Strobel-Verlag. Verfügbar unter: <https://www.ikz.de/uploads/media/1676--Heizungstechnik.pdf>. [Zugriff am 15. Juli 2025].
- [14] BDEW BUNDESVERBAND DER ENERGIE- UND WASSERWIRTSCHAFT E. V. (Herausgeber). *Beheizung des Wohnungsbestandes in Deutschland*. Online. 15. Jan. 2025. Verfügbar unter: <https://www.bdew.de/service/daten-und-grafiken/beheizung-des-wohnungsbestandes-in-deutschland/>. [Zugriff am 15. Juli 2025].
- [15] BDEW BUNDESVERBAND DER ENERGIE- UND WASSERWIRTSCHAFT E. V. (Herausgeber). *Beheizungsstruktur im Neubau in Deutschland – Baugenehmigungen*. Online. 30. Juni 2025. Verfügbar unter: <https://www.bdew.de/service/daten-und-grafiken/beheizungsstruktur-baugenehmigungen-aktuell/>. [Zugriff am 15. Juli 2025].
- [16] BELLMAN, Richard. *Dynamic Programming*. Sixth Printing. Princeton: Princeton University Press, 1972. ISBN 0-691-07951-X. Auch verfügbar unter: <https://gwern.net/doc/statistics/decision/1957-bellman-dynamicprogramming.pdf>. A Rand Corporation Research Study.

- [17] BEZANSON, Jeff; Alan EDELMAN; Stefan KARPINSKI und Viral B. SHAH. *Julia: A Fresh Approach to Numerical Computing*. *SIAM Review*, Bd. 59 (2017), Nr. 1, S. 65–98. Society for Industrial and Applied Mathematics. Verfügbar unter: <https://doi.org/10.1137/141000671>. [Zugriff am 15. Juli 2025]. Auch verfügbar unter: <https://arxiv.org/abs/1411.1607v4> [cs.MS].
- [18] BEZANSON, Jeff; Stefan KARPINSKI; Viral B. SHAH und Alan EDELMAN. *Julia: A Fast Dynamic Language for Technical Computing*. 24. Sep. 2012. Verfügbar unter: <https://arxiv.org/abs/1209.5145v1> [cs.PL]. [Zugriff am 15. Juli 2025].
- [19] BEZANSON, Jeff; Stefan KARPINSKI; Viral B. SHAH und Alan EDELMAN. *Why We Created Julia*. Online. 14. Feb. 2012. Verfügbar unter: <https://julialang.org/blog/2012/02/why-we-created-julia/>. [Zugriff am 15. Juli 2025].
- [20] BOCK, Hans Georg; Ekaterina KOSTINA und Johannes P. SCHLÖDER. Numerical Methods for Parameter Estimation in Nonlinear Differential Algebraic Equations. *GAMM-Mitteilungen*, Bd. 30 (2007), Nr. 2, S. 376–408. Wiley Online Library. Verfügbar unter: <https://doi.org/10.1002/gamm.200790024>. [Zugriff am 15. Juli 2025].
- [21] CELLIER, François E.; Xenofon FLOROS und Ernesto KOFMAN. The Complexity Crisis: Using Modeling and Simulation for System Level Analysis and Design. In: {ÖREN, Tuncer; JANUSZ KACPRZYK; Leifur LEIFSSON; Mohammad S. OBAIDAT und Slawomir KOZIEL (Herausgeber). *Proceedings of the 3rd International Conference on Simulation and Modeling Methodologies, Technologies and Applications [Volume 1: SIMULTECH]*. Setúbal [Portugal]: Science and Technology Publications, 2013. Institute for Systems and Technologies of Information, Control and Communication. ISSN 2184-2841. ISBN 978-989-8565-69-3., S. 5–13. Verfügbar unter: <https://doi.org/10.5220/0004986700050013>. [Zugriff am 15. Juli 2025]. *SIMULTECH 2013*. Reykjavík University, Reykjavík, 29. Juli 2013–31. Juli 2013.
- [22] *Concrete – Density – Heat Capacity – Thermal Conductivity*. Online. 29. Juni 2021. In: Material Properties (Distributor). Verfügbar unter: <https://material-properties.org/concrete-density-heat-capacity-thermal-conductivity/>. [Zugriff am 15. Juli 2025].
- [23] CUSHMANN-ROISIN, Benoit und Jean-Marie BECKERS. *Introduction to Geophysical Fluid Dynamics: Physical and Numerical Aspects*. E-Book. Second Edition. Renata DMOWSKA; Dennis HARTMANN und H. Thomas ROSSBY (Herausgeber). In: International Geophysics Series, Bd. 101. Waltham [Massachusetts]: Academic Press, 2011. ISSN 0074-6142. ISBN 978-0-12-088759-0.
- [24] DALLE, G.; J. SMIT und A. HILL. *Writing your code*. Online. 12. Sep. 2024. Verfügbar unter: <https://modernjuliaworkflows.org/writing/>. [Zugriff am 15. Juli 2025].

- [25] DASSAULT SYSTÈMES AB (Herausgeber). *Dymola: User Manual Volume 1 [Dynamic Modeling Laboratory]*. Version 26. ©2019. Dieses Benutzerhandbuch befindet sich als pdf-Datei im Unterverzeichnis Documentation der Dymola Installation.
- [26] DASSAULT SYSTÈMES AB (Herausgeber). *Dymola: User Manual Volume 2 [Dynamic Modeling Laboratory]*. Version 26. ©2019. Dieses Benutzerhandbuch befindet sich als pdf-Datei im Unterverzeichnis Documentation der Dymola Installation.
- [27] DIN-NORMENAUSSCHUSS HEIZ- UND RAUMLUFTTECHNIK [NHRS]. DIN EN 442-2:2014 (D), *Radiatoren und Konvektoren: Teil 2: Prüfverfahren und Leistungsangabe*. DIN-Norm. Berlin: Beuth Verlag GmbH, 2015. Deutsches Institut für Normung e. V. Verfügbar unter: <https://doi.org/10.31030/2157935>.
- [28] *Druckverlust*. Online. 27. Jan. 2025. In: Wikipedia (Distributor). Wikimedia Foundation. Verfügbar unter: <https://de.wikipedia.org/w/index.php?oldid=252742776>. [Zugriff am 15. Juli 2025].
- [29] ECKL, M. SOPHI Statusmeeting 2018-01-17. Präsentation. 2018. *Arbeitstreffen Förderprojekt SOPHi*. Gebäude F, Hochschule Kaiserslautern, Kaiserslautern, 17. Jan. 2018 09:09:00.
- [30] ECKL, Martin. dynRegelkennlinie.slx. Unveröffentlichtes Simulink-Modell. In: {ECKL, Martin. 4. Juli 2017. *Matlabblock: Dynamische Regelkennlinie*. E-Mail [Berfulicher Austausch].}.
- [31] ECKL, Martin. Erfindung: Dynamische Regelkennlinie zur Erhöhung der Energieeffizienz. Unveröffentlichtes Microsoft Word Dokument. In: {ECKL, Martin. 4. Juli 2017. *Matlabblock: Dynamische Regelkennlinie*. E-Mail [Berfulicher Austausch].}.
- [32] ECKL, Martin; Steven LIU; Vincent BECKER; Sven URSCHEL; Kristina KOWALSKI und Willi GERBIG. BMWi 03ET1300A-D, *Verbundprojekt SOPHi: Systemische Optimierung von Pumpenregelungen zur Steigerung der Energieeffizienz von Heizungs- und Klimaanlageanlagen in der Gebäudetechnik*. Schlussbericht. Forschung für Energieoptimiertes Bauen (EnoB). Hannover: Leibniz-Informationszentrum Technik und Naturwissenschaften und Universitätsbibliothek, 29. März 2019. Bundesministerium für Wirtschaft und Energie. Verfügbar unter: <https://doi.org/10.2314/KXP:1671087518>. [Zugriff am 15. Juli 2025].
- [33] ECKL, Martin; Benedikt MÜLLER; Jochen SCHAAB; Christoph MARK; Steven LIU et al. BMWi 03ET1613A-D, *Verbundprojekt ORSYGET: Entwicklung optimierter Regelungen hydraulischer Systeme in der Gebäudetechnik zur Steigerung der Energieeffizienz von Heizungs- und Klimatisierungssystemen*. Schlussbericht. Forschung für Energieoptimiertes Bauen (EnoB). Hannover: Leibniz-Informationszentrum Technik und Naturwissenschaften und Universitätsbibliothek, 1. Juni 2022. Bundesministerium für Wirtschaft und

Energie. Verfügbar unter: <https://doi.org/10.2314/KXP:185157591X>. [Zugriff am 15. Juli 2025].

- [34] ELMQVIST, Hilding. Modelica Evolution – From My Perspective. In: {TUMMESCHEIT, Hubertus und Karl-Erik ÅRZÉN (Herausgeber). *Book of Abstracts of the 10th International Modelica Conference*. In: {Linköping Electronic Conference Proceedings, Bd. 96}, Nr. 1. Linköping [Schweden]: Modelica Association und Linköping University Electronic Press, 2014. ISSN 1650-3740 [online]. ISSN 1650-3686 [Print]. ISBN 978-91-7519-380-9.}, S. 17–26. Verfügbar unter: <https://doi.org/10.3384/ecp1409617>. [Zugriff am 15. Juli 2025]. *Proceedings of the 10th International Modelica Conference*. Ideon Science Park, Lund [Schweden], 10. März 2014–12. März 2014.
- [35] FRAUNHOFER INSTITUTE FOR SOLAR ENERGY SYSTEMS ISE (Herausgeber). *AI-Controlled Heat Pumps Increase Efficiency*. Pressemitteilung; PDF; online. 17. Dez. 2024. Verfügbar unter: https://www.ise.fraunhofer.de/content/dam/ise/de/documents/presseinformationen/2024/2924_Fraunhofer_ISE_en_PI_AI4HP.pdf. [Zugriff am 15. Juli 2025].
- [36] FRISCHMUTH, Kurt. Mathematische Modellierung eines Heizungskreislaufs: Dynamisierung – Identifikation – Steuerung. Präsentation. 2017. *Forschungskolloquium Numerische Mathematik*. Lehrstuhl für Numerische Mathematik, Universität Rostock, Rostock, 16. Jan. 2017 15:15:00.
- [37] FRITZSON, Peter. *Principles of Object Oriented Modeling and Simulation with Modelica 3.3: A Cyber-Physical Approach*. E-Book. 2nd edition. Piscataway [New Jersey]: IEEE Press, 2015. ISBN 978-1-118-98916-6 [online]. ISBN 978-1-118-85912-4 [Print]. LCCN 2014022271. Verfügbar unter: <https://doi.org/10.1002/9781118989166>.
- [38] *Gaußscher Integralsatz*. Online. 13. Juli 2025. In: Wikipedia (Distributor). Wikimedia Foundation. Verfügbar unter: <https://de.wikipedia.org/w/index.php?oldid=257873984>. [Zugriff am 15. Juli 2025].
- [39] GERBIG, W. und D. WEBERSINKE. *Edapt-Algorithmus für die Drehzahlregelung einer Pumpe in einem Einraum-Heizsystem*. Unveröffentlichtes Paper. 16. Dez. 2016.
- [40] GERBIG, Willi. 4. Statusmeeting zum SOPHi-Projekt. Präsentation. 2017. *Statustreffen zum Projekt “Systemische Optimierung von Pumpenregelungen zur Steigerung der Energieeffizienz von Heizungs- und Klimaanlage der Gebäudetechnik”*. VDMA Haus, Frankfurt am Main, 30. Juni 2017 09:15:00.
- [41] GERDTS, Matthias. *Optimal Control of ODEs and DAEs*. E-Book. 2nd edition. Berlin, Boston: De Gruyter Oldenbourg, 2024. ISBN 978-3-11-079789-3 [online]. ISBN 978-3-11-079769-5 [Print]. Verfügbar unter: <https://doi.org/10.1515/9783110797893>.

- [42] GLASBERG BAUSTOFFE GMBH (Herausgeber). *Greenshield leichter Alaun mit Isolierung: Verpackung und Verbraucherinformationen*. Online. © [ca. 2020]. Verfügbar unter: http://www.glasbergbaustoffe.de/de/i_urunler_sap.php. [Zugriff am 23. Okt. 2023].
- [43] GOODFELLOW, Ian; Yoshua BENGIO und Aaron COURVILLE. *Deep Learning*. First Edition. Adaptive Computation and Machine Learning. Cambridge [Massachusetts]: The MIT Press, 2016. ISBN 978-0-262-03561-3. LCCN 2016022992. Auch verfügbar unter: <http://www.deeplearningbook.org>.
- [44] GRAMMLING, Franz. *Rechnergestützte Analyse von Heizungsrohrnetzen*. Dissertation. Stuttgart: Universität Stuttgart, Institut für Kernenergietechnik und Energiesysteme, 6. Mai 1988. Standort: Technische Informationsbibliothek, Universitätsbibliothek Hannover, Hannover. ISSN 0173-6892.
- [45] GROLLIUS, Horst-W. *Grundlagen der Hydraulik*. 7., aktualisierte Auflage. München: Carl Hanser Verlag GmbH & Co. KG, 2015. ISBN 978-3-446-44104-0 [online]. ISBN 978-3-446-44275-7 [Print]. Auch verfügbar unter: <https://doi.org/10.3139/9783446441040>.
- [46] GRUNDFOS A/S. *Verfahren zum Steuern einer drehregelbaren Heizungsumwälzpumpe*. Europäisches Patent. Carsten SKOVMOSE KALLESOE; Pierre VADSTRUP; Erik OLSEN; Per BARTH; Lars R. ENEVOLDSEN und Nils BIDSTRUP (Erfinder). Anmeldung: 11. Dez. 2002. Veröffentlichungstag: 19. Jan. 2011. Int Cl.: F24D 19/10 (2006.01); F04D 15/00 (2006.01). EP 1 323 986 B1. Verfügbar unter: Google Patents (Distributor), <https://patents.google.com/patent/EP1323986B1/>. [Zugriff am 15. Juli 2025].
- [47] GÜLICH, Johann Friedrich. *Kreiselpumpen: Handbuch für Entwicklung, Anlagenplanung und Betrieb*. E-Book. 4., aktualisierte und erweiterte Auflage. Berlin: Springer Vieweg, 2013. ISBN 978-3-642-40032-2 [online]. ISBN 978-3-642-40031-5 [Print]. Verfügbar unter: <https://doi.org/10.1007/978-3-642-40032-2>.
- [48] HAASE, Helmut; Heyno GARBE und Hendrik GERTH. *Grundlagen der Elektrotechnik*. 3. erweiterte Auflage. Hannover: SchöneworthVerlag, 2009. ISBN 3-9808805-5-9.
- [49] HAGMANN, Gert. *Grundlagen der Elektrotechnik: Das bewährte Lehrbuch für Studierende der Elektrotechnik und anderer technischer Studiengänge ab 1. Semester*. 8., korrigierte Auflage. Wiebelsheim [Rheinland-Pfalz]: AULA-Verlag GmbH, 2001. ISBN 3-89104-661-8.
- [50] HAMKINS, Christopher P. Präsentation KSB für Statusmeeting „SOPHI“. Präsentation. 2017. *Statustreffen zum Projekt "Systemische Optimierung von Pumpenregelungen zur Steigerung der Energieeffizienz von Heizungs- und Klimaanlage der Gebäudetechnik" (SOPHI)*. Berufsbildungszentrum von KSB, Frankenthal [Rheinland-Pfalz], 19. Jan. 2017 09:13:00.

- [51] HEATON, Jeff. *Introduction to Neural Networks with Java*. Second Edition. St. Louis [Missouri]: Heaton Research, 2008. ISBN 978-1-60439-008-7.
- [52] *Heizkessel*. Online. 22. Juni 2025. In: Wikipedia (Distributor). Wikimedia Foundation. Verfügbar unter: <https://de.wikipedia.org/w/index.php?oldid=257257269>. [Zugriff am 15. Juli 2025].
- [53] HENNESSEY, John L. und David A. PATTERSON. *Rechnerarchitektur: Analyse, Entwurf, Implementierung, Bewertung*. 1. Auflage. Wiesbaden: Springer Fachmedien, 1994. ISBN 978-3-663-13953-9 [online]. ISBN 978-3-528-05173-0 [Print]. Auch verfügbar unter: <https://doi.org/10.1007/978-3-663-13953-9>.
- [54] HORLACHER, Hans-Burkhard und Ulf HELBIG (Herausgeber). *Rohrleitungen: Grundlagen, Rohrwerkstoffe, Komponenten*. E-Book. 2. Auflage. Springer Reference Technik, Bd. 1. Berlin: Springer Vieweg, 2016. ISBN 978-3-642-39782-0 [online]. ISBN 978-3-642-39781-3 [Print]. Verfügbar unter: <https://doi.org/10.1007/978-3-642-39782-0>.
- [55] HTFLUX ENGINEERING GMBH (Herausgeber). *Der Wärmeübergangswiderstand*. Online. 26. Juni 2020. Verfügbar unter: <https://www.htflux.com/de/documentation/randbedingungen/waermeuebergangswiderstand/>. [Zugriff am 15. Juli 2025].
- [56] IDELCHIK, I. E. *Flow Resistance: A Design Guide for Engineers*. First Edition. Erwin FRIED (Herausgeber). Philadelphia: Taylor & Francis, 1989. ISBN 978-0-203-75575-4 [online]. ISBN 1-56032-487-2 [Print]. Auch verfügbar unter: <https://doi.org/10.1201/9780203755754>.
- [57] INTEGRATED ENVIRONMENTAL SOLUTIONS [IES] (Herausgeber). *Table 6 Thermal Conductivity, Specific Heat Capacity and Density*. Online. ©2021. Verfügbar unter: https://help.iesve.com/ve2021/table_6_thermal_conductivity_specific_heat_capacity_and_density.htm. [Zugriff am 15. Juli 2025].
- [58] *Interpolation (Mathematik)*. Online. 17. März 2025. In: Wikipedia (Distributor). Wikimedia Foundation. Verfügbar unter: <https://de.wikipedia.org/w/index.php?oldid=254284186>. [Zugriff am 15. Juli 2025].
- [59] *Introducing Julia: Types*. Online. 15. Mai 2025. In: Wikibooks (Distributor). Wikimedia Foundation. Verfügbar unter: <https://en.wikibooks.org/w/index.php?oldid=4491967>. [Zugriff am 15. Juli 2025].
- [60] ISMAILOV, Vugar E. On the approximation by neural networks with bounded number of neurons in hidden layers. *Journal of Mathematical Analysis and Applications*, Bd. 417 (2014), Nr. 2, S. 963–969. Elsevier. ISSN 0022-247X. Verfügbar unter: <https://doi.org/10.1016/j.jmaa.2014.03.092>. [Zugriff am 15. Juli 2025].

- [61] JOG, C. S. *Continuum Mechanics: Foundations and Applications of Mechanics*. E-Book. Third Edition. Cambridge IISc Series, Bd. 1. Delhi: Cambridge University Press, 2015. ISBN 978-1-316-13405-4 [online]. ISBN 978-1-107-09135-1 [Print]. LCCN 2015001499. Verfügbar unter: <https://doi.org/10.1017/CB09781316134054>.
- [62] JOG, C. S. *Fluid Mechanics: Foundations and Applications of Mechanics*. E-Book. Third Edition. Cambridge IISc Series, Bd. 2. Delhi: Cambridge University Press, 2015. ISBN 978-1-316-13403-0 [online]. ISBN 978-1-107-09129-0 [Print]. LCCN 2014048627. Verfügbar unter: <https://doi.org/10.1017/CB09781316134030>.
- [63] *Julia (Programmiersprache)*. Online. 26. Mai 2025. In: Wikipedia (Distributor). Wikimedia Foundation. Verfügbar unter: <https://de.wikipedia.org/w/index.php?oldid=256361371>. [Zugriff am 15. Juli 2025].
- [64] JULIALANG.ORG CONTRIBUTORS (Herausgeber). *benchmarks.csv*. CSV-Datei. ©2020. Verfügbar unter: <https://github.com/JuliaLang/Microbenchmarks/blob/a963d284b09d04b3e0374f6dd46ec4b039ed5569/bin/benchmarks.csv>. [Zugriff am 15. Juli 2025].
- [65] JULIALANG.ORG CONTRIBUTORS (Herausgeber). *Julia Micro-Benchmarks*. Online. ©2020. Verfügbar unter: <https://julia-lang.org/benchmarks/>. [Zugriff am 15. Juli 2025].
- [66] JULIALANG.ORG CONTRIBUTORS (Herausgeber). *Producing the Julia Microbenchmarks plot*. Jupyter Notebook. ©2020. Verfügbar unter: <https://github.com/JuliaLang/Microbenchmarks/blob/a963d284b09d04b3e0374f6dd46ec4b039ed5569/bin/benchmarks.ipynb>. [Zugriff am 15. Juli 2025].
- [67] KAYNAKLI, O. A study on residential heating energy requirement and optimum insulation thickness. *Renewable Energy: Official Journal of the World Renewable Energy Network*, Bd. 33 (2008), Nr. 6, S. 1164–1172. Elsevier. ISSN 0960-1481. Verfügbar unter: <https://doi.org/10.1016/j.renene.2007.07.001>. [Zugriff am 15. Juli 2025].
- [68] KINGMA, Diederik P. und Jimmy Lei BA. *Adam: A Method for Stochastic Optimization*. 30. Jan. 2017. Verfügbar unter: <https://arxiv.org/abs/1412.6980v9> [cs.LG]. [Zugriff am 15. Juli 2025]. Auch verfügbar unter: <https://doi.org/10.48550/arXiv.1412.6980>.
- [69] KIRCHHOFF, Studiosus. Ueber den Durchgang eines elektrischen Stromes durch eine Ebene, insbesondere durch eine kreisförmige. *Annalen der Physik und Chemie*, Bd. 140 (1845), Nr. 4, S. 497–514. Wiley Online Library. Verfügbar unter: <https://doi.org/10.1002/andp.18451400402>. [Zugriff am 29. Sep. 2019]. Alternative Zählung: Band 64 der Poggendorffs Annalen.

- [70] KIRK, Donald E. *Optimal Control Theory: An Introduction*. Dover Edition. Mineola [New York]: Dover Publications, 2004. ISBN 978-0-486-43484-1. LCCN 2003070111.
- [71] KNECHT, Detlev (Chefredakteur). Was ist eigentlich der Unterschied... zwischen einer Pumpenkennlinie und einer Rohrnetzkenlinie. Online. *IKZ-Praxis: für die SHK-Haustechnik*, 2001, Nr. 7, S. 5. Strobel-Verlag. Verfügbar unter: <https://www.ikz.de/ikz-praxis-archiv/p0107/010705.php>. [Zugriff am 15. Juli 2025].
- [72] KNEUSEL, Ronald T. *Practical Deep Learning: A Python-Based Introduction*. E-Book. First Edition. San Francisco: No Starch Press, 2021. ISBN 978-1-7185-0075-4. LCCN 2020035098.
- [73] *Kontinuumsmechanik*. Online. 16. Juni 2025. In: Wikipedia (Distributor). Wikimedia Foundation. Verfügbar unter: <https://de.wikipedia.org/w/index.php?oldid=257045425>. [Zugriff am 15. Juli 2025].
- [74] KOWALSKI, Kristina. Statusmeeting SOPHI. Präsentation. 2017. *Statustreffen zum Projekt "Systemische Optimierung von Pumpenregelungen zur Steigerung der Energieeffizienz von Heizungs- und Klimaanlageanlagen der Gebäudetechnik"*. VDMA Haus, Frankfurt am Main, 30. Juni 2017 09:15:00.
- [75] KOWALSKI, Kristina. Statusmeeting SOPHi. Präsentation. 2016. *Arbeitstreffen Förderprojekt SOPHi*. Berufsbildungszentrum von KSB, Frankenthal [Rheinland-Pfalz], 8. März 2016.
- [76] KRAL, Christian. *Modelica: Objektorientierte Modellbildung von Drehfeldmaschinen*. E-Book. 1. Auflage. München: Carl Hanser Verlag GmbH & Co. KG, 2019. ISBN 978-3-446-45733-1 [online]. ISBN 978-3-446-45551-1 [Print]. Verfügbar unter: <https://doi.org/10.3139/9783446457331>.
- [77] KRESS, K. Nr. 205705-AS7-1, *Vergleichsprüfung Energieeffizienzindex*. Prüfbericht. Zertifikat ID. 40041138. 13. Feb. 2015. VDE Prüf- und Zertifizierungsinstitut GmbH. Archivierte Kopie verfügbar unter: Internet Archive (Distributor), <https://web.archive.org/web/20150418101606/http://de.grundfos.com/content/dam/dach/produkte/heizungsumwaelzpumpen/ALPHA2-VDE-Zertifikat.pdf>. [Zugriff am 15. Juli 2025].
- [78] KROLKIEWICZ, Hans Jürgen. *Günstig bauen*. E-Book. 3. Auflage. Meine Immobilie. München: Rudolf Haufe Verlag GmbH & Co. KG, 2010. ISBN 978-3-448-10165-2. Verfügbar unter: https://www.wiso-net.de/document/HAUF_9783448101652287. [Zugriff am 15. Juli 2025].
- [79] KSB SE & Co. KGAA (Herausgeber). *Anlagenkennlinie*. Online. [ca. 2017]. Verfügbar unter: <https://www.ksb.com/de-global/kreiselpumpenlexikon/artikel/anlagenkennlinie-1074942>. [Zugriff am 15. Juli 2025].

- [80] KSB SE & Co. KGAA. *Verfahren zur Regelung einer drehzahlvariablen Umwälzpumpe sowie Umwälzpumpe*. Deutsches Patent. Martin Eckl (Erfinder). Anmeldung: 3. März 2017. Veröffentlichungstag: 6. Sep. 2018. Int Cl.: F04D 15/00 (2006.01). DE 10 2017 203 474 A1. Verfügbar unter: Google Patents (Distributor), <https://patents.google.com/patent/DE102017203474A1>. [Zugriff am 15. Juli 2025].
- [81] KSB SE & Co. KGAA (Herausgeber). *Was sind die erweiterten Funktionen der Calio?* Online. 15. Aug. 2024. Verfügbar unter: <https://support.ksb.com/hc/de/articles/23202862621331-Was-sind-die-erweiterten-Funktionen-der-Calio>. [Zugriff am 15. Juli 2025].
- [82] KUNDE, Jeannette. *Thermostat: Die Zahlen und ihre Bedeutung*. Online. 24. Juli 2024. Verfügbar unter: <https://heizung.de/heizung/wissen/thermostat-die-zahlen-und-ihre-bedeutung/>. [Zugriff am 15. Juli 2025].
- [83] LAASCH, Thomas und Erhard LAASCH. *Haustechnik: Grundlagen – Planung – Ausführung*. E-Book. 13., durchgesehen und aktualisierte Auflage. Wiesbaden: Springer Vieweg, 2013. ISBN 978-3-8348-9900-2 [online]. ISBN 978-3-8348-1260-5 [Print]. Verfügbar unter: <https://doi.org/10.1007/978-3-8348-9900-2>.
- [84] LAWRENCE BERKELEY NATIONAL LABORATORY (Herausgeber). *modelica-buildings/Buildings/Fluid/HeatExchangers/Radiators/RadiatorEN442_2.mo*. Online. 21. Apr. 2017. Verfügbar unter: https://github.com/lbl-srg/modelica-buildings/blob/940031ca15a74aa3cce6ffa095c9fdf212592a4d/Buildings/Fluid/HeatExchangers/Radiators/RadiatorEN442_2.mo#L13-L14. [Zugriff am 15. Juli 2025].
- [85] LITTMAN, Michael L. *The Reward Hypothesis*. Video; online. In: WHITE, Martha und Adam WHITE. *Fundamentals of Reinforcement Learning*. In: {Reinforcement Learning Specialization [4 Kurse], Kurs 1}, Modul 3, Video 4. 2019. Brown University. Verfügbar unter: Coursera (Distributor), <https://www.coursera.org/lecture/fundamentals-of-reinforcement-learning/michael-littman-the-reward-hypothesis-q6x0e>. [Zugriff am 15. Juli 2025].
- [86] LOBIANCO, Antonello. *Julia Quick Syntax Reference: A Pocket Guide for Data Science Programming*. E-Book. First Edition. New York: Apress Media LLC, 2019. ISBN 978-1-4842-5190-4 [online]. ISBN 978-1-4842-5189-8 [Print]. Verfügbar unter: <https://doi.org/10.1007/978-1-4842-5190-4>.
- [87] LOETI. *Erklär mal: Geregelter- und Ungeregelter Pumpen*. Online. Gentner Verlag GmbH, 29. Juli 2011. In: SBZ Monteur (Distributor). Verfügbar unter: <https://www.sbz-monteur.de/erklaer-mal/erklaer-mal-geregelte-und-ungeregelte-pumpen>. [Zugriff am 15. Juli 2025].
- [88] *Logarithmisches Mittel*. Online. 10. März 2025. In: Wikipedia (Distributor). Wikimedia Foundation. Verfügbar unter: <https://de.wikipedia.org/w/index.php?oldid=254058113>. [Zugriff am 15. Juli 2025].

- [89] MARTÍNEZ, Rubén Mulero; Beñat Arregi GOIKOLEA; Iñigo Mendialdua BERTIA und Roberto Garay MARTÍNEZ. Designing a generalised reward for Building Energy Management Reinforcement Learning agents. In: {2021 6th International Conference on Smart and Sustainable Technologies [SpliTech 2021]. Split [Kroatien]: Institute of Electrical and Electronics Engineers, 2021. ISBN 978-953-290-112-2 [online]. ISBN 978-1-6654-4202-2 [Print].}, S. 346–351. Verfügbar unter: <https://doi.org/10.23919/SpliTech52315.2021.9566345>. [Zugriff am 15. Juli 2025]. 2021 6th International Conference on Smart and Sustainable Technologies. Bluesun Hotel Elaphusa, Bol [Kroatien], 8. Sep. 2021–11. Sep. 2021.
- [90] *Modelica*. Online. 23. Mai 2025. In: Wikipedia (Distributor). Wikimedia Foundation. Verfügbar unter: <https://en.wikipedia.org/w/index.php?oldid=1291781876>. [Zugriff am 15. Juli 2025].
- [91] MODELICA ASSOCIATION (Herausgeber). *Modelica Tools*. Online. [ca. 1999]. Verfügbar unter: <https://www.modelica.org/tools>. [Zugriff am 15. Juli 2025].
- [92] MODELICA ASSOCIATION (Herausgeber). *Modelica®– A Unified Object-Oriented Language for Systems Modeling: Language Specification*. PDF. Version 3.6. 9. März 2023. Verfügbar unter: <https://specification.modelica.org/maint/3.6/MLS.pdf>. [Zugriff am 17. Juli 2025].
- [93] MODELICA ASSOCIATION (Herausgeber). *ModelicaStandardLibrary/Modelica/Fluid/Interfaces.mo*. Online. 1. Juli 2019. Verfügbar unter: <https://github.com/modelica/ModelicaStandardLibrary/blob/e531f9a959e3bc02a021c49ed65839622fbdbb6f/Modelica/Fluid/Interfaces.mo#L7-L22>. [Zugriff am 15. Juli 2025].
- [94] MODELICA ASSOCIATION (Herausgeber). *ModelicaStandardLibrary/Modelica/Fluid/Machines.mo*. Online. 6. Aug. 2019. Verfügbar unter: <https://github.com/modelica/ModelicaStandardLibrary/blob/e531f9a959e3bc02a021c49ed65839622fbdbb6f/Modelica/Fluid/Machines.mo#L464-L465>. [Zugriff am 15. Juli 2025].
- [95] MODELICA ASSOCIATION (Herausgeber). *ModelicaStandardLibrary/Modelica/Fluid/Machines.mo*. Online. 7. Jan. 2020. Verfügbar unter: <https://github.com/modelica/ModelicaStandardLibrary/blob/d6b0767046127d3277a366766e1da0b0445ae525/Modelica/Fluid/Machines.mo#L426-L429>. [Zugriff am 15. Juli 2025].
- [96] MODELICA ASSOCIATION (Herausgeber). *ModelicaStandardLibrary/Modelica/Fluid/Pipes.mo*. Online. 18. Nov. 2019. Verfügbar unter: <https://github.com/modelica/ModelicaStandardLibrary/blob/e531f9a959e3bc02a021c49ed65839622fbdbb6f/Modelica/Fluid/Pipes.mo#L2884-L2920>. [Zugriff am 15. Juli 2025].

- [97] MODELICA ASSOCIATION (Herausgeber). *ModelicaStandardLibrary/Modelica/Fluid/Vessels.mo*. Online. 1. Juli 2019. Verfügbar unter: <https://github.com/modelica/ModelicaStandardLibrary/blob/e531f9a959e3bc02a021c49ed65839622fbdbb6f/Modelica/Fluid/Vessels.mo#L87-L96>. [Zugriff am 15. Juli 2025].
- [98] MODELICA ASSOCIATION (Herausgeber). *ModelicaStandardLibrary/Modelica/Media/Water/package.mo*. Online. 8. Nov. 2019. Verfügbar unter: <https://github.com/modelica/ModelicaStandardLibrary/blob/e531f9a959e3bc02a021c49ed65839622fbdbb6f/Modelica/Media/Water/package.mo#L37-L68>. [Zugriff am 15. Juli 2025].
- [99] MURPHY, Sean Patrick. *The Rise of Data Products*. Vortrag, 2013. *Mid-Maryland Data Science Meetup*. Johns Hopkins University Applied Physics Laboratory, Laurel [US-MD], 29. Jan. 2013 18:00:00. Archivierte Kopie verfügbar unter: Internet Archive (Distributor), <https://web.archive.org/web/20130206120736/http://www.datacommunitydc.org/blog/2013/01/the-rise-of-data-products>. [Zugriff am 15. Juli 2025].
- [100] MUSCHEVICI, Radu; Alex POTANIN; Ewan TEMPERO und James NOBLE. Multiple Dispatch in Practice. In: {HARRIS, Gail E.; Gregor KICZALES; Dirk RIEHLE und Andrew P. BLACK (Herausgeber). *Conference Proceedings: 23rd Annual ACM Conference on Object-Oriented Programming, Systems, Languages, and Applications*. New York: Association for Computing Machinery, 2008. ISBN 978-1-60558-215-3.}, S. 563–582. Verfügbar unter: <https://doi.org/10.1145/1449764.1449808>. [Zugriff am 15. Juli 2025]. Auch verfügbar unter: <https://potanin.github.io/files/MuscheviciPotaninTemperoNobleOOPSLA2008.pdf>. *OOPSLA 2008*. Nashville Convention Center, Nashville [US-TN], 19. Okt. 2008–23. Okt. 2008.
- [101] NAKAYAMA, Yasuki. *Introduction to Fluid Mechanics*. E-Book. Second Edition. Oxford: Butterworth-Heinemann, 2018. ISBN 978-0-08-102438-6 [online]. ISBN 978-0-08-102437-9 [Print]. Verfügbar unter: <https://www.sciencedirect.com/book/9780081024379/introduction-to-fluid-mechanics>.
- [102] NEUKIRCHEN, Florian (Herausgeber). *Die Folgen des Klimawandels*. E-Book. 1. Auflage. Berlin: Springer, 2019. ISBN 978-3-662-59581-7 [online]. ISBN 978-3-662-59580-0 [Print]. Verfügbar unter: <https://doi.org/10.1007/978-3-662-59581-7>.
- [103] NG, Andrew. *Backpropagation Intuition (Optional)*. Video; online. In: Ng, Andrew; Kian KATANFOROOSH und Younes Bensouda MOURRI. *Neural Networks and Deep Learning*. In: {Deep Learning Specialization [5 Kurse], Kurs 1}, Modul 3, Video 10. 2017. DeepLearning.AI. Verfügbar unter: Coursera (Distributor), <https://www.coursera.org/lecture/neural-networks-deep-learning/backpropagation-intuition-optional-6dDj7>. [Zugriff am 15. Juli 2025]. Auch verfügbar unter: <https://www.youtube.com/watch?v=yXcQ4B-YSjQ>.

- [104] NG, Andrew. *Computing a Neural Network's Output*. Video; online. In: NG, Andrew; Kian KATANFOROOSH und Younes Bensouda MOURRI. *Neural Networks and Deep Learning*. In: {Deep Learning Specialization [5 Kurse], Kurs 1}, Modul 3, Video 3. 2017. DeepLearning.AI. Verfügbar unter: Coursera (Distributor), <https://www.coursera.org/lecture/neural-networks-deep-learning/computing-a-neural-networks-output-tyAGh>. [Zugriff am 15. Juli 2025]. Auch verfügbar unter: <https://www.youtube.com/watch?v=rM0drD61IoU>.
- [105] NG, Andrew. *Gradient Descent for Neural Networks*. Video; online. In: NG, Andrew; Kian KATANFOROOSH und Younes Bensouda MOURRI. *Neural Networks and Deep Learning*. In: {Deep Learning Specialization [5 Kurse], Kurs 1}, Modul 3, Video 9. 2017. DeepLearning.AI. Verfügbar unter: Coursera (Distributor), <https://www.coursera.org/lecture/neural-networks-deep-learning/gradient-descent-for-neural-networks-Wh8NI>. [Zugriff am 15. Juli 2025]. Auch verfügbar unter: https://www.youtube.com/watch?v=7bLEWDZng_M.
- [106] NG, Andrew. *Vectorizing Across Multiple Examples*. Video; online. In: NG, Andrew; Kian KATANFOROOSH und Younes Bensouda MOURRI. *Neural Networks and Deep Learning*. In: {Deep Learning Specialization [5 Kurse], Kurs 1}, Modul 3, Video 4. 2017. DeepLearning.AI. Verfügbar unter: Coursera (Distributor), <https://www.coursera.org/lecture/neural-networks-deep-learning/vectorizing-across-multiple-examples-ZCcMM>. [Zugriff am 15. Juli 2025]. Auch verfügbar unter: <https://www.youtube.com/watch?v=xy5M0Qpx3aQ>.
- [107] NIELSEN, Michael A. *Neural Networks and Deep Learning*. First Edition. San Francisco: Determination Press, 2015. Auch verfügbar unter: <http://neuralnetworksanddeeplearning.com/index.html>.
- [108] NISHIKAWA, Hiroaki. *Accurate Piecewise Linear Continuous Approximations to One-Dimensional Curves: Error Estimates and Algorithms*. Preprint. Jan. 1998. Verfügbar unter: https://www.researchgate.net/publication/265146527_Accurate_Piecewise_Linear_Continuous_Approximations_to_One-Dimensional_Curves_Error_Estimates_and_Algorithms. [Zugriff am 15. Juli 2025].
- [109] *Number of users of smart homes in Germany from 2019 to 2028 (in millions)*. Diagramm; online. 25. Sep. 2023. In: Statista (Distributor). Statista Research Department. Verfügbar unter: <https://www.statista.com/forecasts/480442/number-of-smart-homes-in-the-smart-home-market-in-germany>. [Zugriff am 15. Juli 2025].
- [110] NYTSCH-GEUSEN, Christoph; Jörg HUBER; Manuel LJUBIJANKIC und Jörg RÄDLER. Modelica BuildingSystems: eine Modellbibliothek zur Simulation komplexer energietechnischer Gebäudesysteme. *Bauphysik*, Bd. 35 (2013), Nr. 1, S. 21–29. Ernst & Sohn. Verfügbar unter: <https://doi.org/10.1002/bapi.201310045>. Siehe auch: <https://github.com/UdK-VPT/BuildingSystems>. [Zugriff am 15. Juli 2025].

- [111] PAOLUCCI, S. *Continuum Mechanics and Thermodynamics of Matter*. E-Book. 1st edition. New York: Cambridge University Press, 2016. ISBN 978-1-316-10616-7 [online]. ISBN 978-1-107-08995-2 [Print]. LCCN 2015034141. Verfügbar unter: <https://doi.org/10.1017/CB09781316106167>.
- [112] PEIRELINCK, Thijs; Frederik RUELENS und Geert DECNONINCK. Using Reinforcement Learning for Optimizing Heat Pump Control in a Building Model in Mod-*elica*. In: {2018 IEEE International Energy Conference [ENERGYCON 2018]. Piscataway [New Jersey]: Institute of Electrical and Electronics Engineers, 2018. ISBN 978-1-5386-3669-5 [online]. ISBN 978-1-5386-1283-5 [Print].}, S. 579–584. Verfügbar unter: <https://doi.org/10.1109/ENERGYCON.2018.8398832>. [Zugriff am 15. Juli 2025]. 2018 IEEE International Energy Conference. St. Raphael Resort, Limassol [Zypern], 3. Juli 2018–7. Juli 2018.
- [113] POWELL, James. Scientists Reach 100% Consensus on Anthropogenic Global Warming. *Bulletin of Science, Technology & Society*, Bd. 37 (2019), Nr. 4, S. 183–184. SAGE Publications. Verfügbar unter: <https://doi.org/10.1177/0270467619886266>. [Zugriff am 15. Juli 2025].
- [114] PRÖHL, Lukas; Harald ASCHEMANN; Kristina KOWALSKI und Frank-Hendrik WURM. Energy-Optimal Control for a Heating Circulator. In: {*Proceedings: IECON 2020 The 46th Annual Conference of the IEEE Industrial Electronics Society*. Piscataway [New Jersey]: Institute of Electrical and Electronics Engineers, 18. Nov. 2020. Nanyang Technological University und IEEE Industrial Electronics Society. ISSN 2577-1647 [online]. ISSN 1553-572X [Print]. ISBN 978-1-7281-5414-5 [online]. ISBN 978-1-7281-5415-2 [Print].}, S. 1980–1985. Verfügbar unter: <https://doi.org/10.1109/IECON43393.2020.9254363>. [Zugriff am 15. Juli 2025]. *IECON 2020 The 46th Annual Conference of the IEEE Industrial Electronics Society*. Virtual, 19. Sep. 2020–21. Sep. 2020.
- [115] REUTER ONLINESHOP GMBH (Herausgeber). *Strahlungswärme vs. Konvektionswärme: alles eine Frage der Behaglichkeit*. Online. ©2004. Verfügbar unter: <https://www.reuter.de/magazin/strahlungswaerme-vs-konvektionswaerme.html>. [Zugriff am 15. Juli 2025].
- [116] ROBINSON, Reece (Technischer Content-Manager). L-CBS-WP-001 04-19, *Pumping Control Methods and Their Impact on System Efficiency*. White Paper. Brookshire [Texas]: Grundfos Americas, 27. Apr. 2019. Grundfos Building Services. Verfügbar unter: <https://www.esmagazine.com/articles/99473-pumping-control-methods-and-their-impact-on-system-efficiency-white-paper>. [Zugriff am 29. Mai 2021].
- [117] SCHIPPERGES, Michael. Texte 103/2020, *Umwelteinstellungen in Deutschland von 1971 bis 2019: Zeitreihenanalyse anhand externer Datenquellen*. Forschungsbericht. Forschungskennzahl 3717 16 103 0, FB000149/ZW,3. Dessau-Roßlau [Sachsen-Anhalt]: Umweltbundesamt, Juni 2020. sociodimensions – Institute for Socio-cultural Research. ISSN 1862-4804. Verfügbar

- unter: <https://www.umweltbundesamt.de/publikationen/umwelteinstellungen-in-deutschland-von-1971-bis>. [Zugriff am 15. Juli 2025].
- [118] SCHLOBACH, Martin. *Wie funktioniert ein Thermostatventil?* Online. 7. Nov. 2024. Verfügbar unter: <https://www.haustechnikverstehen.de/thermostatheizung/>. [Zugriff am 15. Juli 2025].
- [119] SCHULZ, Gerd. *Regelungstechnik 1: Lineare und Nichtlineare Regelung, Rechnergestützter Reglerentwurf*. E-Book. 4., überarbeitete Auflage. München: Oldenbourg Wissenschaftsverlag GmbH, 2010. ISBN 978-3-486-70794-6 [online]. ISBN 978-3-486-59194-1 [Print]. Verfügbar unter: <https://doi.org/10.1524/9783486707946>.
- [120] SENGUPTA, Avik. *Julia High Performance: Optimizations, distributed computing, multithreading, and GPU programming with Julia 1.0 and beyond*. E-Book. Second Edition. Birmingham: Packt Publishing Ltd. 2019. ISBN 978-1-78829-811-7.
- [121] SHEELA, K. Gnana und S. N. DEEPA. Review on Methods to Fix Number of Hidden Neurons in Neural Networks. *Mathematical Problems in Engineering*, Bd. 2013 (2013), Nr. 1, Article ID 425740. Hindawi Publishing Corporation. ISSN 1563-5147 [online]. ISSN 1024-123X [Print]. Verfügbar unter: <https://doi.org/10.1155/2013/425740>. [Zugriff am 15. Juli 2025].
- [122] SPITZENVERBAND DER GEBÄUDETECHNIK (Herausgeber). *Optimierung von Heizungsanlagen im Bestand: Fachregel*. Broschüre. Version 1.2. 2016. Archivierte Kopie verfügbar unter: Internet Archive (Distributor), https://web.archive.org/web/20170809000107/https://www.vdzev.de/wp-content/uploads/2014/02/Fachregel_hydraulischer-Abgleich.pdf. [Zugriff am 15. Juli 2025].
- [123] STOCKER, Anita (Chefredakteurin). Pump it up. *test*, 2018, Nr. 5, S. 63–67. Stiftung Warentest. Verfügbar unter: <https://www.test.de/Heizungspumpen-im-Test-1567473-0/>. [Zugriff am 15. Juli 2025].
- [124] SUTTON, Richard S. und Andrew G. BARTO. *Reinforcement Learning: An Introduction*. E-Book. Second Edition. Adaptive Computation and Machine Learning. Cambridge [Massachusetts]: The MIT Press, 2018. ISBN 978-0-262-03924-6. LCCN 2018023826. Verfügbar unter: <http://www.incompleteideas.net/book/RLbook2020.pdf>.
- [125] SWAMEE, Prabhata K. und Akalank K. JAIN. Explicit equations for pipe-flow problems. *Journal of the Hydraulics Division*, Bd. 102 (1976), Nr. 5, S. 657–664. American Society of Civil Engineers. Verfügbar unter: https://www.researchgate.net/publication/280018838_Explicit_eqations_for_pipe-flow_problems. [Zugriff am 15. Juli 2025].

- [126] TEICHMANN, Martin (Verantwortlicher für den Inhalt). *Die Pumpenkennlinie: Kennzahl und Funktion*. Online. Sascha CAMILL (Co-Betreiber). [ca. 2016]. Verfügbar unter: <https://www.kesselheld.de/pumpenkennlinie/>. [Zugriff am 15. Juli 2025].
- [127] TEICHMANN, Martin (Verantwortlicher für den Inhalt). *Gasheizung: Modell Vergleich, Preise, Förderung & Funktion*. Online. Sascha CAMILL (Co-Betreiber). [ca. 2016]. Verfügbar unter: <https://www.kesselheld.de/gasheizung/>. [Zugriff am 15. Juli 2025].
- [128] THE JULIA PROJECT (Herausgeber). *The Julia Language*. Version 1.3.1. 4. Aug. 2019. Verfügbar unter: <https://raw.githubusercontent.com/JuliaLang/docs.julialang.org/assets/julia-1.3.1.pdf>. [Zugriff am 15. Juli 2025].
- [129] THEODOR HEIMEIER METALLWERK GMBH (Herausgeber). *Thermostat-Ventilunterteile: für alle Thermostat-Köpfe und Stellantriebe*. Prospekt. Version 1.2. 2010. Verfügbar unter: http://heizungsfachshop.de/media/pdf/1e/50/3d/prospekt_heimeier_thermostatventil.pdf. [Zugriff am 7. Nov. 2019].
- [130] THERMTEST INC. (Herausgeber). *potassium alum*. Online. In: Materials Database. © 2023. Der Eintrag für „potassium alum“ kann über das Suchfeld am schnellsten gefunden werden. Verfügbar unter: <https://thermtest.com/thermal-resources/materials-database>. [Zugriff am 15. Juli 2025].
- [131] WÄCHTER, A. und L. BIEGLER. On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming. *Mathematical Programming*, Series A, Bd. 106 (2006), Nr. 1, S. 25–57. Springer Science+Business Media. ISSN 1563-5147 [online]. ISSN 1436-4646 [Print]. Verfügbar unter: <https://doi.org/10.1007/s10107-004-0559-y>. [Zugriff am 15. Juli 2025].
- [132] WAGNER, W. und A. PRUSS. The IAPWS Formulation 1995 for the Thermodynamic Properties of Ordinary Water Substance for General and Scientific Use. *Journal of Physical and Chemical Reference Data*, Bd. 31 (2002), Nr. 2, S. 387–535. AIP Publishing. ISSN 1529-7845 [online]. ISSN 0047-2689 [Print]. Verfügbar unter: <https://doi.org/10.1063/1.1461829>. [Zugriff am 15. Juli 2025].
- [133] WAN, Yi; Abhishek NAIK und Richard S. SUTTON. Learning and Planning in Average-Reward Markov Decision Processes. In: {MEILA, Marina und Tong ZHANG (Herausgeber). *Proceedings of the 38th International Conference on Machine Learning*. Proceedings of Machine Learning Research, 2021., S. 10653–10662. Verfügbar unter: <https://proceedings.mlr.press/v139/wan21a/wan21a.pdf>. [Zugriff am 15. Juli 2025]. Auch verfügbar unter: <https://arxiv.org/abs/2006.16318v3> [cs.LG]. *Volume 139: International Conference on Machine Learning*. Virtual, 18. Juli 2021–24. Juli 2021.

- [134] WEBERSINKE, Danilo. Research on algorithm for night setback control and auto-adapt algorithm for circulators: Generation of an algorithm for auto adapt control (eAdapt). Präsentation. 2015. *Final Meeting of the Project „Autoadapt für Heizungspumpen“*. Xylem Montecchio Factory, Montecchio Maggiore [Italien], 17. Feb. 2015.
- [135] WERFF, Ellen van der; Linda STEG und Angela RUEPERT. My company is green, so am I: the relationship between perceived environmental responsibility of organisations and government, environmental self-identity, and pro-environmental behaviours. *Energy Efficiency*, Bd. 14 (2021), Nr. 50. Springer Nature B.V. ISSN 1570-6478 [online]. ISSN 1570-646X [Print]. Verfügbar unter: <https://doi.org/10.1007/s12053-021-09958-9>. [Zugriff am 15. Juli 2025].
- [136] WESCHE, Wolfgang. *Radiale Kreiselpumpen: Berechnung und Konstruktion der hydrodynamischen Komponenten*. E-Book. 2., aktualisierte und erweiterte Auflage. Berlin: Springer Vieweg, 2016. ISBN 978-3-662-48912-3 [online]. ISBN 978-3-662-48911-6 [Print]. Verfügbar unter: <https://doi.org/10.1007/978-3-662-48912-3>.
- [137] WETTER, Michael; Wangda Zuo; Thierry S. NOUIDUI und Xiufeng PANG. Modelica Buildings library. *Journal of Building Performance Simulation: Cross-industry Multi-domain Modelling Language Applications for Building Simulation*, Bd. 7 (2014), Nr. 4, S. 253–270. Taylor & Francis Online. Verfügbar unter: <https://doi.org/10.1080/19401493.2013.765506>. Siehe auch: <https://github.com/lbl-srg/modelica-buildings>. [Zugriff am 19. Mai 2022].
- [138] WHITE, Adam. *Specifying Policies*. Video; online. In: WHITE, Martha und Adam WHITE. Fundamentals of Reinforcement Learning. In: {Reinforcement Learning Specialization [4 Kurse], Kurs 1}, Modul 4, Video 1. 2019. University of Alberta und Alberta Machine Intelligence Institute. Verfügbar unter: Coursera (Distributor), <https://www.coursera.org/lecture/fundamentals-of-reinforcement-learning/specifying-policies-SsygZ>. [Zugriff am 15. Juli 2025].
- [139] WHITE, Martha. *Optimal Policies*. Video; online. In: WHITE, Martha und Adam WHITE. Fundamentals of Reinforcement Learning. In: {Reinforcement Learning Specialization [4 Kurse], Kurs 1}, Modul 4, Video 7. 2019. University of Alberta und Alberta Machine Intelligence Institute. Verfügbar unter: Coursera (Distributor), <https://www.coursera.org/lecture/fundamentals-of-reinforcement-learning/optimal-policies-AjTR1>. [Zugriff am 15. Juli 2025].
- [140] WHITE, Martha. *Using Optimal Value Functions to Get Optimal Policies*. Video; online. In: WHITE, Martha und Adam WHITE. Fundamentals of Reinforcement Learning. In: {Reinforcement Learning Specialization [4 Kurse], Kurs 1}, Modul 4, Video 8. 2019. University of Alberta und Alberta Machine Intelligence Institute. Verfügbar unter: Coursera (Distributor), <https://www.coursera.org/lecture/fundamentals-of-reinforcement-learning/using-optimal-value-functions-to-get-optimal-policies-93kAE>. [Zugriff am 15. Juli 2025].

- [141] WHITE, Martha. *Week 2 Summary*. Video; online. In: WHITE, Martha und Adam WHITE. Fundamentals of Reinforcement Learning. In: {Reinforcement Learning Specialization [4 Kurse], Kurs 1}, Modul 3, Video 7. 2019. University of Alberta und Alberta Machine Intelligence Institute. Verfügbar unter: Coursera (Distributor), <https://www.coursera.org/lecture/fundamentals-of-reinforcement-learning/week-2-summary-qlFDv>. [Zugriff am 15. Juli 2025].
- [142] WHITE, Martha und Adam WHITE. *Course Introduction*. Video; online. In: WHITE, Martha und Adam WHITE. Fundamentals of Reinforcement Learning. In: {Reinforcement Learning Specialization [4 Kurse], Kurs 1}, Modul 1, Video 2. 2019. University of Alberta und Alberta Machine Intelligence Institute. Verfügbar unter: Coursera (Distributor), <https://www.coursera.org/lecture/fundamentals-of-reinforcement-learning/course-introduction-gY00M>. [Zugriff am 15. Juli 2025].
- [143] XPRODUCTS DEUTSCHLAND GMBH (Herausgeber). *Xaggregate*. Online. 2016. Verfügbar unter: <https://www.xprds.de/cubeportfolio/xaggregate/>. [Zugriff am 15. Juli 2025].
- [144] XYLEM INC. *How to choose the correct speed control for heating systems in building services applications*. White Paper; PDF. Rye Brook [New York]: Xylem Inc., März 2017. Xylem Applied Water Systems. Verfügbar unter: <https://www.xylem.com/siteassets/support/white-papers/white-paper-resources/correct-speed-control-for-heating-systems-in-bs.pdf>.

Anhang A

Abbildungen

A.1 Simuliertes Nutzerverhalten

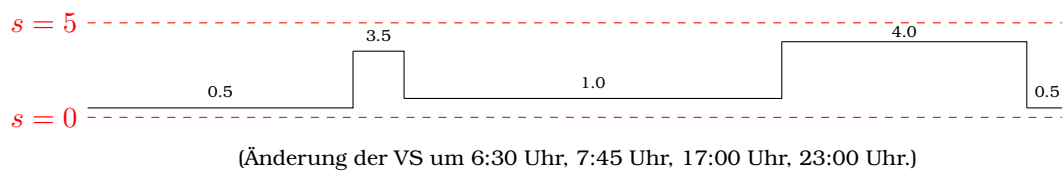


Abb. A.1: 1. Tagesablauf: Arbeitstag, Variante 1

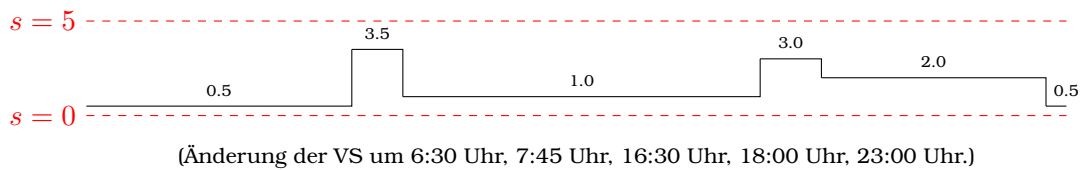


Abb. A.2: 2. Tagesablauf: Arbeitstag, Variante 2

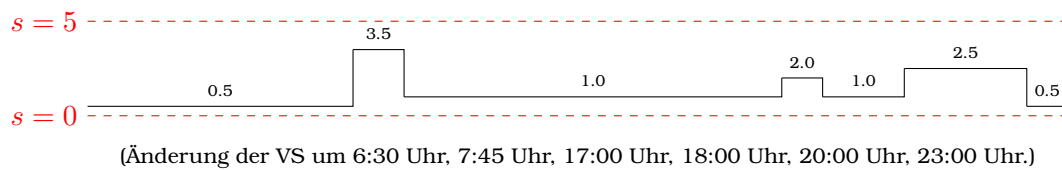


Abb. A.3: 3. Tagesablauf: Arbeitstag, Variante 3

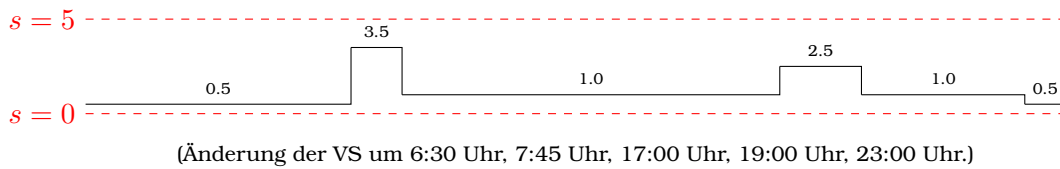


Abb. A.4: 4. Tagesablauf: Arbeitstag, Variante 4

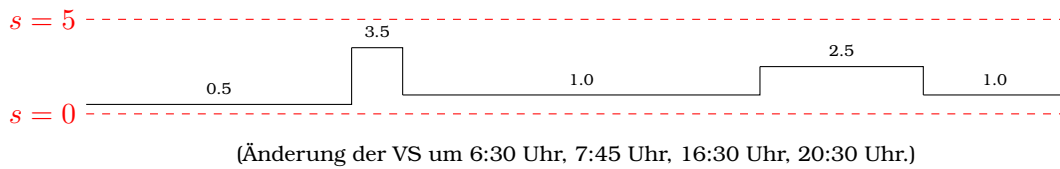


Abb. A.5: 5. Tagesablauf: Arbeitstag, Variante 5

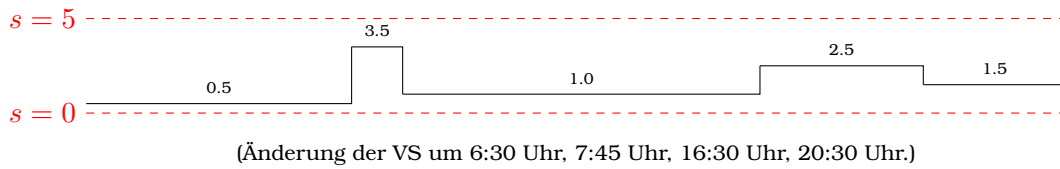


Abb. A.6: 6. Tagesablauf: Arbeitstag Freitag, Variante 1

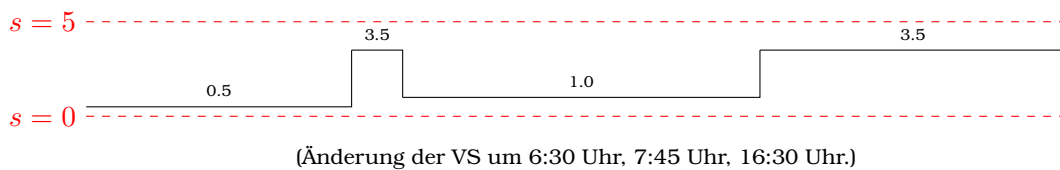
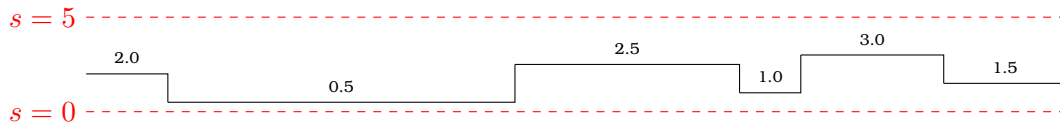


Abb. A.7: 7. Tagesablauf: Arbeitstag Freitag, Variante 2



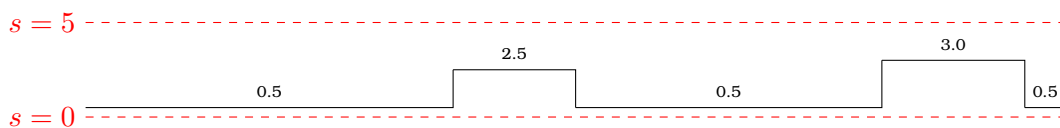
(Änderung der VS um 2:00 Uhr, 10:30 Uhr, 16:00 Uhr, 17:30 Uhr, 21:00 Uhr.)

Abb. A.8: 8. Tagesablauf: Wochenendtag, Variante 1



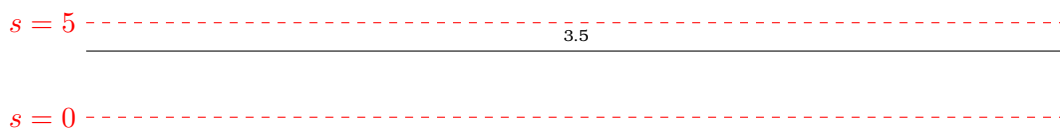
(Änderung der VS um 12:30 Uhr, 15:00 Uhr, 19:00 Uhr, 23:00 Uhr.)

Abb. A.9: 9. Tagesablauf: Wochenendtag, Variante 2



(Änderung der VS um 9:00 Uhr, 12:00 Uhr, 19:30 Uhr, 23:30 Uhr.)

Abb. A.10: 10. Tagesablauf: Wochenendtag, Variante 3



(Keine Änderung der VS.)

Abb. A.11: 11. Tagesablauf: Tag mit konstanter Ventilstellung

A.2 Heizkurve

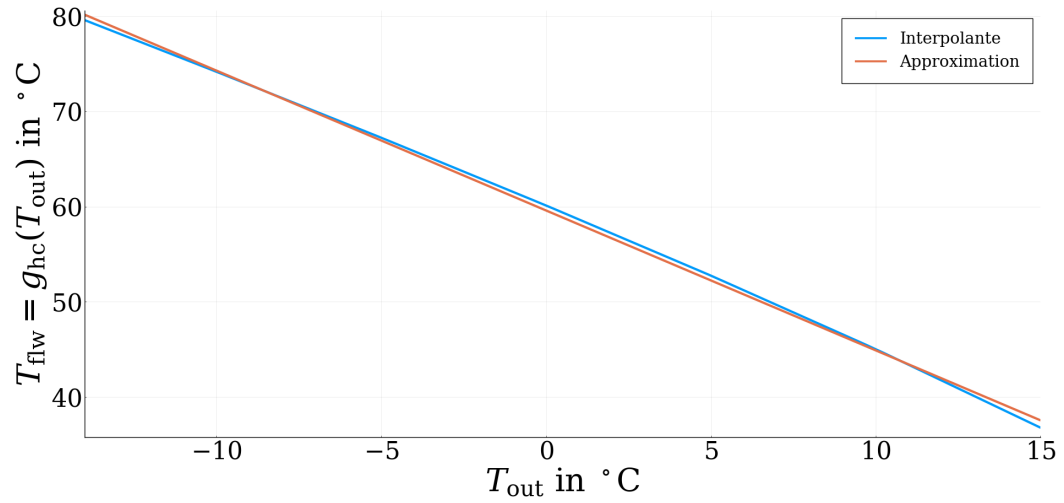


Abb. A.12: Heizkurve

A.3 Ventilkennlinie

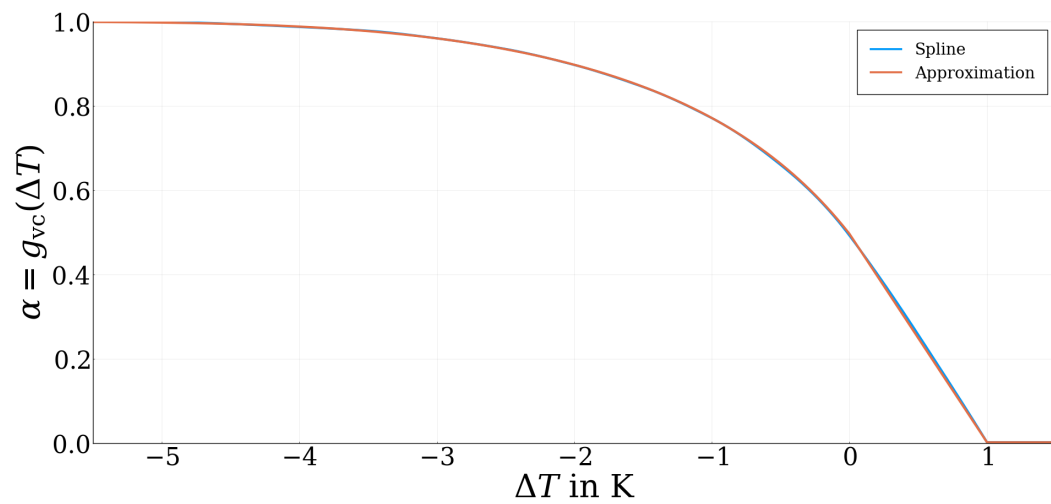


Abb. A.13: Ventilkennlinie

A.5 Benchmark-Modell in Dymola

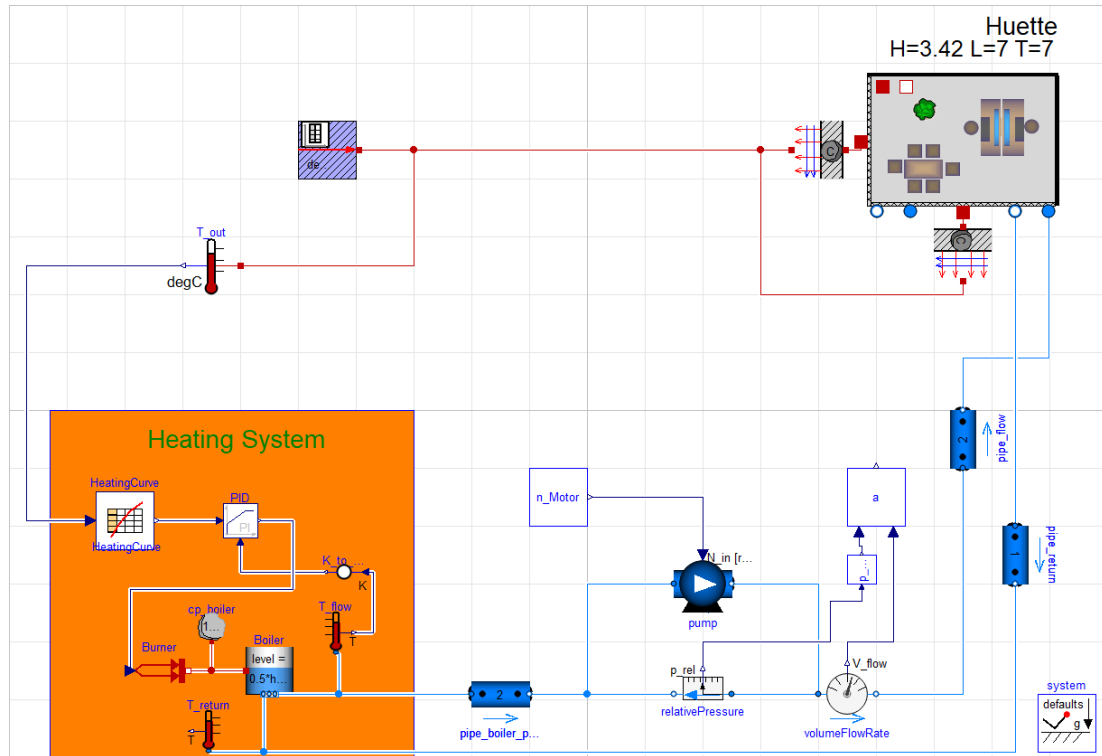


Abb. A.15: Benchmark-Modell in Dymola [75, Folie 5 f.]

Anhang B

Tabellen

B.1 Heizkurve

T_{out}	T_{flw}
-14.0 °C	80.49 °C
-10.0 °C	75.00 °C
-5.0 °C	68.01 °C
0.0 °C	60.85 °C
5.0 °C	53.46 °C
10.0 °C	45.76 °C
15.0 °C	37.58 °C

Tab. B.1: Werte der Heizkurve des Benchmark-Modells

Die Tabelle B.1 zeigt die Wertepaare für die Werte der Außentemperatur T_{out} und der zugehörigen Vorlauftemperatur T_{flw} , aus denen sich die Heizkurve des Benchmark-Modells bildet.

B.2 Ventilkennlinien

Die Tabelle B.2 zeigt die Massenströme der Ventilkennlinien des Heimeier V-exakt Thermostat-Ventilunterteils für die sechs verschiedenen Voreinstellungen (VE 1 – VE 6) in Abhängigkeit der Regeldifferenz ΔT bei einem Differenzdruck von $\Delta p = 10 \text{ kPa} = 0.1 \text{ bar}$.

ΔT	VE 1	VE 2	VE 3	VE 4	VE 5	VE 6
1.0 K	0.00 $\frac{\text{kg}}{\text{h}}$	0.00 $\frac{\text{kg}}{\text{h}}$	0.00 $\frac{\text{kg}}{\text{h}}$	0.00 $\frac{\text{kg}}{\text{h}}$	0.00 $\frac{\text{kg}}{\text{h}}$	0.00 $\frac{\text{kg}}{\text{h}}$
0.5 K	10.05 $\frac{\text{kg}}{\text{h}}$	17.12 $\frac{\text{kg}}{\text{h}}$	25.54 $\frac{\text{kg}}{\text{h}}$	31.79 $\frac{\text{kg}}{\text{h}}$	38.04 $\frac{\text{kg}}{\text{h}}$	44.57 $\frac{\text{kg}}{\text{h}}$
0.0 K	15.22 $\frac{\text{kg}}{\text{h}}$	27.17 $\frac{\text{kg}}{\text{h}}$	42.39 $\frac{\text{kg}}{\text{h}}$	58.24 $\frac{\text{kg}}{\text{h}}$	72.73 $\frac{\text{kg}}{\text{h}}$	87.50 $\frac{\text{kg}}{\text{h}}$
-0.5 K	16.85 $\frac{\text{kg}}{\text{h}}$	31.25 $\frac{\text{kg}}{\text{h}}$	49.73 $\frac{\text{kg}}{\text{h}}$	71.02 $\frac{\text{kg}}{\text{h}}$	97.44 $\frac{\text{kg}}{\text{h}}$	121.54 $\frac{\text{kg}}{\text{h}}$
-1.0 K	17.39 $\frac{\text{kg}}{\text{h}}$	31.52 $\frac{\text{kg}}{\text{h}}$	51.14 $\frac{\text{kg}}{\text{h}}$	75.00 $\frac{\text{kg}}{\text{h}}$	113.83 $\frac{\text{kg}}{\text{h}}$	148.14 $\frac{\text{kg}}{\text{h}}$
-1.5 K	17.66 $\frac{\text{kg}}{\text{h}}$	32.07 $\frac{\text{kg}}{\text{h}}$	51.99 $\frac{\text{kg}}{\text{h}}$	76.14 $\frac{\text{kg}}{\text{h}}$	124.74 $\frac{\text{kg}}{\text{h}}$	172.16 $\frac{\text{kg}}{\text{h}}$
-2.0 K	17.93 $\frac{\text{kg}}{\text{h}}$	32.61 $\frac{\text{kg}}{\text{h}}$	52.27 $\frac{\text{kg}}{\text{h}}$	76.99 $\frac{\text{kg}}{\text{h}}$	132.45 $\frac{\text{kg}}{\text{h}}$	190.63 $\frac{\text{kg}}{\text{h}}$
-2.5 K	18.21 $\frac{\text{kg}}{\text{h}}$	32.88 $\frac{\text{kg}}{\text{h}}$	53.13 $\frac{\text{kg}}{\text{h}}$	77.56 $\frac{\text{kg}}{\text{h}}$	138.03 $\frac{\text{kg}}{\text{h}}$	204.89 $\frac{\text{kg}}{\text{h}}$
-3.0 K	18.75 $\frac{\text{kg}}{\text{h}}$	33.42 $\frac{\text{kg}}{\text{h}}$	53.41 $\frac{\text{kg}}{\text{h}}$	78.41 $\frac{\text{kg}}{\text{h}}$	141.76 $\frac{\text{kg}}{\text{h}}$	214.40 $\frac{\text{kg}}{\text{h}}$
-3.5 K	19.29 $\frac{\text{kg}}{\text{h}}$	33.70 $\frac{\text{kg}}{\text{h}}$	54.26 $\frac{\text{kg}}{\text{h}}$	79.26 $\frac{\text{kg}}{\text{h}}$	144.41 $\frac{\text{kg}}{\text{h}}$	220.92 $\frac{\text{kg}}{\text{h}}$
-4.0 K	19.57 $\frac{\text{kg}}{\text{h}}$	34.51 $\frac{\text{kg}}{\text{h}}$	54.83 $\frac{\text{kg}}{\text{h}}$	79.83 $\frac{\text{kg}}{\text{h}}$	145.74 $\frac{\text{kg}}{\text{h}}$	225.27 $\frac{\text{kg}}{\text{h}}$
-4.5 K	19.84 $\frac{\text{kg}}{\text{h}}$	34.78 $\frac{\text{kg}}{\text{h}}$	55.40 $\frac{\text{kg}}{\text{h}}$	80.68 $\frac{\text{kg}}{\text{h}}$	146.81 $\frac{\text{kg}}{\text{h}}$	228.26 $\frac{\text{kg}}{\text{h}}$
-5.0 K	20.38 $\frac{\text{kg}}{\text{h}}$	35.05 $\frac{\text{kg}}{\text{h}}$	55.97 $\frac{\text{kg}}{\text{h}}$	81.53 $\frac{\text{kg}}{\text{h}}$	147.61 $\frac{\text{kg}}{\text{h}}$	230.43 $\frac{\text{kg}}{\text{h}}$

Tab. B.2: Ventilkennlinien des V-exakt Thermostat-Ventilunterteils

Anhang C

Parametrierung

C.1 Physikalische Konstanten

$$\begin{aligned} c_p &= 4184 \frac{\text{J}}{\text{kg}\cdot\text{K}}, & c_V &= 4184 \frac{\text{J}}{\text{kg}\cdot\text{K}}, & \rho_w &= 995.586 \frac{\text{kg}}{\text{m}^3}, & \eta_w &= 0.001 \text{ Pa}\cdot\text{s}, \\ \lambda_w &= 0.598 \frac{\text{W}}{\text{m}\cdot\text{K}}, & M_w &= 0.018015268 \frac{\text{kg}}{\text{mol}}, & g &= 9.80665 \frac{\text{m}}{\text{s}^2}, & T_0 &= 273.15 \text{ K} \end{aligned}$$

C.2 Normbedingungen für Radiatoren und Konvektoren

$$T_{\text{flw,nom}} = 75^\circ\text{C}, \quad T_{\text{rtn,nom}} = 65^\circ\text{C}, \quad T_{\text{room,nom}} = 20^\circ\text{C}, \quad p_{\text{air,nom}} = 101.325 \text{ kPa}$$

C.3 Parametrierung des DGL-Systems

$$\begin{aligned} n_{\text{htr}} &\approx 1.3000000, \\ C_{\text{room}} &\approx 2.0279199 \cdot 10^5 \frac{\text{kgm}^2}{\text{s}^2\text{K}}, & C_{\text{wall}} &\approx 1.1558757 \cdot 10^7 \frac{\text{kgm}^2}{\text{s}^2\text{K}}, \\ C_{\text{fc}} &\approx 1.5162511 \cdot 10^7 \frac{\text{kgm}^2}{\text{s}^2\text{K}}, & C_{\text{htr}} &\approx 2.0827659 \cdot 10^{-1} \frac{\text{kgm}^2}{\text{s}^2\text{K}}, \\ \nu_{\text{wall,int}} &\approx 2.3313751 \cdot 10^2 \frac{\text{kgm}^2}{\text{s}^3\text{K}}, & \nu_{\text{wall,ext}} &\approx 5.5993200 \cdot 10^1 \frac{\text{kgm}^2}{\text{s}^3\text{K}}, \\ \nu_{\text{fc,int}} &\approx 4.8571576 \cdot 10^2 \frac{\text{kgm}^2}{\text{s}^3\text{K}}, & \nu_{\text{win}} &\approx 4.1520000 \frac{\text{kgm}^2}{\text{s}^3\text{K}}, \\ \nu_{\text{htr}} &\approx 1.4204189 \cdot 10^1 \frac{\text{kgm}^2}{\text{s}^3\text{K}}, & \nu_w &\approx 4.1655318 \cdot 10^6 \frac{\text{kg}}{\text{ms}^2\text{K}}, \\ m_{\text{hc}} &\approx -1.4698850, & T_{\text{hc},0} &\approx 7.3423193 \cdot 10^2 \text{ K}, \\ c_{\text{vc},3} &\approx 2.1291703 \cdot 10^{-2} \frac{1}{\text{K}^3}, & c_{\text{vc},2} &\approx 5.6815221 \cdot 10^{-2} \frac{1}{\text{K}^2}, \\ c_{\text{vc},1} &\approx 8.2493665 \cdot 10^{-1} \frac{1}{\text{K}}, & c_{\text{vc},0} &\approx -6.8665007 \cdot 10^{-1}, \\ \alpha_{\text{min}} &\approx 2.5770612 \cdot 10^{-3} \end{aligned}$$

C.4 Anfangsbedingungen des DGL-Systems

$$\begin{aligned} T_{\text{room},0} &= 295.19661 \text{ K}, & T_{\text{wall},0} &= 291.89729 \text{ K}, & T_{\text{fc},0} &= 293.54211 \text{ K}, \\ T_{\text{htr},0} &= 319.96447 \text{ K}, & T_{\text{rtn},0} &= 315.72938 \text{ K}, & T_{\text{flw},0} &= 325.36872 \text{ K}, \\ \alpha_0 &= 0.47370648, & \dot{V}_0 &= 4.0932226 \cdot 10^{-5} \frac{\text{m}^3}{\text{s}} \end{aligned}$$

$$\begin{aligned} n_{\text{pump},0} &:= n_{\text{pump}}(t_0) = 3500 \frac{1}{\text{min}} \\ t_0 &= 0 \text{ s}, \quad t_f = 90 \text{ d} = 7776000 \text{ s} \end{aligned}$$

C.5 Parametrierung für die Systemidentifikation

$$\begin{aligned} t_0 &= 0 \text{ s}, \quad t_f^{\text{id}} = 7 \text{ d} = 604800 \text{ s}, \\ \Delta\tau^{\text{id}} &= 24 \text{ h} = 86400 \text{ s}, \quad \Delta t_\mu^{\text{id}} = 1 \text{ h} = 3600 \text{ s}. \end{aligned}$$

C.6 Parametrierung für die Lösung nach Bellman

$$\Delta t^{\text{Bm}} = 1 \text{ h} = 3600 \text{ s} \qquad \Delta n^{\text{Bm}} = 1 \frac{1}{\text{min}}$$

C.7 Parametrierung der elementaren Steuerstrategien

C.7.1 Regelung mit konstanter Drehzahl

$$o_{\text{dsgn}} = 0.15 \qquad n^{\text{cs}} = 3464.5896 \frac{1}{\text{min}}$$

C.7.2 Konstantdruckregelung

$$o_{\text{dsgn}} = 0.15 \qquad H^{\text{cp}} = 4.4036001 \text{ m}$$

C.7.3 Proportionaldruckregelung

$$o_{\text{dsgn}} = 0.15 \qquad m^{\text{pp}} = 39121.997 \frac{\text{s}}{\text{m}^2} \qquad H^{\text{pp}} = 2.7000069 \text{ m}$$

C.7.4 Regelung auf einer quadratischen Regelkurve

$$o_{\text{dsgn}} = 0.15 \quad a^{\text{qP}} = 2.3222987 \cdot 10^9 \frac{\text{s}^2}{\text{m}^5} \quad H^{\text{qP}} = 0.0 \text{ m}$$

C.8 Parametrierung von Grundfos' Autoadapt

$$\Delta t^{\text{Gf}} = 4 \text{ h} = 14400 \text{ s}$$

$$\begin{aligned} \sigma_{\dot{V}} &= 1.0 & \dot{V}_{G_{\text{min}},0} &= 2.35 \cdot 10^{-5} \frac{\text{m}^3}{\text{s}} & q^{\text{Gf}} &= 0.35 \\ \sigma_H &= 1.0 & \dot{V}_{G_{\text{max}},0} &= 6.5 \cdot 10^{-5} \frac{\text{m}^3}{\text{s}} & \Delta \dot{V}^{\text{Gf}} &= 0.30 \cdot 10^{-5} \frac{\text{m}^3}{\text{s}} \end{aligned}$$

C.9 Parametrierung von KSB's Dynamic Control

$$\Delta t^{\text{KSB}} = 3 \text{ h} = 10800 \text{ s} \quad \Delta \dot{V}^{\text{KSB}} = 1.0 \cdot 10^{-6} \frac{\text{m}^3}{\text{s}} \quad n^{\text{KSB}} = 0.0092596879 \frac{1}{\text{min}}$$

$$u_{\text{stat}}^{\text{KSB}}(t) = u^{\text{cP}}(t) \quad \text{mit } H^{\text{cP}} = H_{\text{cp}}^{\text{KSB}} = 4.4036001 \text{ m}$$

$$u_{\text{stat}}^{\text{KSB}}(t) = u^{\text{pP}}(t) \quad \text{mit } m^{\text{pP}} = m_{\text{pp}}^{\text{KSB}} = 39121.997 \frac{\text{s}}{\text{m}^2} \quad \text{und } H^{\text{pP}} = H_{\text{pp}}^{\text{KSB}} = 2.7000069 \text{ m}$$

C.10 Parametrierung der KI-Steuerstrategie

$$\begin{aligned} l_0 &= 0.26769970 & l_1 &= 0.53962828 \\ \alpha_1 &= 0.44732449 & \alpha_2 &= 0.49674087 & \alpha_3 &= 0.53633681 & \alpha_4 &= 0.66296755 \\ R_{\text{max}} &= 1.0 & \beta_R &= 0.5 \end{aligned}$$

$$\Delta n_{\text{min}}^{\text{AI}} = -200 \frac{1}{\text{min}} \quad \Delta n_{\text{max}}^{\text{AI}} = 200 \frac{1}{\text{min}}$$

List. C.1: Julia-Code des Zufallsgenerators für die Zufallsvariablen X_ε und X_A

```
using Random
using Distributions

global rng = Xoshiro( 245993)
```

$$\Delta t_{\text{mxmn}}^{\text{AI}} = 6 \text{ d} = 518400 \text{ s} \quad \Delta t_{\text{trans}}^{\text{AI}} = 2 \text{ h} = 7200 \text{ s} \quad \Delta t_{\text{tp}}^{\text{AI}} = 3 \text{ h} = 10800 \text{ s}$$

$$K_{\text{tp}}^{\text{AI}} = 9 \quad J_{\text{tp}}^{\text{AI}} = 6 \quad \alpha_{\text{tp}}^{\text{AI}} = 0.007 \quad \beta_{\text{tp}}^{\text{AI}} = 0.0075$$

$$C_{\text{tp}}^{\text{AI}} = \begin{pmatrix} 5 & -5 & 8 & -8 & 3 & -3 \\ 9 & -9 & -2 & 2 & 6 & -6 \\ 1 & -1 & 7 & -7 & 4 & -4 \\ 6 & -6 & 9 & -9 & 2 & -2 \\ 4 & -4 & -1 & 1 & 7 & -7 \\ 8 & -8 & 3 & -3 & 5 & -5 \\ 2 & -2 & -6 & 6 & 9 & -9 \\ 7 & -7 & 4 & -4 & 1 & -1 \\ 3 & -3 & -5 & 5 & 8 & -8 \\ -5 & 5 & -8 & 8 & -3 & 3 \end{pmatrix}$$

$$L = 3 \quad n^{[0]} = 3 \quad n^{[1]} = 4 \quad n^{[2]} = 6 \quad n^{[3]} = 1$$

$$\sigma^{[1]}(x) = \max(0, x) \quad \sigma^{[2]}(x) = \max(0, x) \quad \sigma^{[3]}(x) = x$$

$$\mathbf{W}^{[1]} = \begin{pmatrix} -2.6428585 & 0.0 & -0.35205057 \\ 2.2786865 & 0.0 & 0.30392069 \\ 0.87966216 & 0.0 & 0.11724597 \\ -1.8094405 & 0.0 & -0.24149616 \end{pmatrix}$$

$$\mathbf{b}^{[1]} = (-0.91053724 \quad 0.33962083 \quad 1.1694673 \quad 1.0012451)^{\text{T}}$$

$$\mathbf{W}^{[2]} = \begin{pmatrix} -0.28875834 & 1.3394264 & -0.47399265 & 1.1127292 \\ -0.33333752 & 2.1034744 & -0.81492561 & 0.93043774 \\ -1.2250060 & 0.90235049 & -0.21582907 & 0.084793746 \\ 0.78961825 & 0.48418134 & -1.4711024 & 0.68672162 \\ 0.85810739 & 0.80973327 & -1.2864505 & 1.2013204 \\ 0.26236266 & 1.0823262 & -1.8749444 & -1.4620082 \end{pmatrix}$$

$$\mathbf{b}^{[2]} = (-1.9343761 \quad -1.5099860 \quad -0.14537634$$

$$\quad -0.68253255 \quad 0.24412383 \quad 0.88668257)^{\text{T}}$$

$$\mathbf{W}^{[3]} = (-0.59904778 \quad -0.76424080 \quad -1.2484338$$

$$\quad -0.24746658 \quad -0.69205534 \quad 0.17439552)$$

$$\mathbf{b}^{[3]} = (0.15104280)$$

$$\Delta t^{\text{AI}} = 3 \text{ h} = 10800 \text{ s} \quad \alpha_{\text{ctrl}}^{\text{AI}} = 0.00015 \quad \beta_{\text{ctrl}}^{\text{AI}} = 0.0005 \quad \varepsilon_{\text{ctrl}}^{\text{AI}} = 0.015$$

Anhang D

Ausgewählte Quelltexte des Julia-Modells

D.1 Berechnung des Betriebspunktes der Pumpe

List. D.1: Julia-Quelltext von g_opp.jl

```
using Polynomials
using ForwardDiff

using PumpConstantsModule

function g_opp( n_pump::Number, alpha::Number)

    atol =  $\dot{V}_{s\_ref}[\text{end}] * 1e-8$ 
    r = n_pump / n_ref

    i = 2
    while i < length( $\dot{V}_{s\_ref}$ ) &&
        H_ref_etp(  $\dot{V}_{s\_ref}[i]$ ) >  $1/r^2 * H_{sys}( \alpha, r*\dot{V}_{s\_ref}[i]$ )
        i += 1
    end
    c3, c2, c1, c0 = GetCoeffs( i-1, H_ref_etp)
    a1 =  $1/r * (128 * (\eta_w * l_{pipe}) / (\pi * \rho_w * g * d_{pipe}^4))$ 
    a2 =  $\Delta p_{ref} / (g * k_{vs}^2 * \alpha^2 * \rho_{ref})$ 

     $\dot{V}_{s\_tmp}$  = RealRootsOfCubicPolynomial( c3, c2 - a2, c1 - a1, c0)
    for  $\dot{V}_{s\_tmp}$  in  $\dot{V}_{s\_tmp}$ 
        if  $\dot{V}_{s\_ref}[i-1]$  <  $\dot{V}_{s\_tmp}$  ≤  $\dot{V}_{s\_ref}[i]$ 
            return r *  $\dot{V}_{s\_tmp}$ 
        end
        if abs(  $\dot{V}_{s\_ref}[i-1]$  -  $\dot{V}_{s\_tmp}$ ) < atol
            return r *  $\dot{V}_{s\_ref}[i-1]$ 
        end
    end
end

end
```

D.2 Berechnung der Modellparameter

List. D.2: Julia-Quelltext von BuildingConstantsModule.jl

```
module BuildingConstantsModule

# Konstanten für das Einraum-Modell
export H_room, L_room, B_room, H_win1, B_win1, H_win2, B_win2
export A_win1, A_win2, A_wall1, A_wall2, A_wall3, A_wall4, A_floor, A_ceiling,
      A_win, A_wall, A_fc
export d_c1, d_c2, d_c3, d_c4, d_f1, d_f2, d_f3, d_f4, d_f5, d_f6,
      d_w1, d_w2, d_w3, d_w4, d_w5, d_eff
export λ_c1, λ_c2, λ_c3, λ_c4, λ_f1, λ_f2, λ_f3, λ_f4, λ_f5, λ_f6,
      λ_w1, λ_w2, λ_w3, λ_w4, λ_w5
export c_c1, c_c2, c_f1, c_f2, c_w1, c_w2, c_air
export ρ_c1, ρ_c2, ρ_f1, ρ_f2, ρ_w1, ρ_w2, ρ_air
export h_c_in, h_c_out, h_f_in, h_w_in, h_w_out
export U_win, U_c, U_c_int, U_c_ext, U_f, U_f_int, U_f_ext,
      U_fc, U_fc_int, U_fc_ext, U_wall, U_wall_int, U_wall_ext

# WillisRealHut_N2:

# RAUM- UND FENSTERMASSE
#
# Höhe des Raumes [H_room] = m
const H_room = 3.42
# Länge des Raumes [L_room] = m
const L_room = 7.0
# Breite des Raumes [B_room] = m
const B_room = 7.0
# Höhe des 1. Fensters [H_win1] = m
const H_win1 = 0.8
# Breite des 1. Fensters [B_win1] = m
const B_win1 = 0.5
# Höhe des 2. Fensters [H_win2] = m
const H_win2 = 0.8
# Breite des 2. Fensters [B_win2] = m
const B_win2 = 1.0

# FLÄCHEN:
#
# Fläche des 1. Fensters [A_win1] = m2
const A_win1 = H_win1 * B_win1
# Fläche des 2. Fensters [A_win2] = m2
const A_win2 = H_win2 * B_win2
# Fläche der 1. Wand [A_wall1] = m2
const A_wall1 = H_room * L_room - A_win1
# Fläche der 2. Wand [A_wall2] = m2
const A_wall2 = H_room * B_room - A_win2
# Fläche der 3. Wand [A_wall3] = m2
const A_wall3 = H_room * L_room
# Fläche der 4. Wand [A_wall4] = m2
const A_wall4 = H_room * B_room
```

```

# Fläche des Bodens [A_floor] = m2
const A_floor = L_room * B_room
# Fläche der Decke [A_ceiling] = m2
const A_ceiling = L_room * B_room

# Fläche aller Fenster [A_win] = m2
const A_win = A_win1 + A_win2
# Fläche aller Wände [A_wall] = m2
const A_wall = A_wall1 + A_wall2 + A_wall3 + A_wall4
# Gemittelte Fläche von Boden und Decke [A_fc] = m2
const A_fc = 1/2 * (A_floor + A_ceiling)

# DECKE:
#
# Dicke der 1. Schicht der Decke [d_c1] = m
# siehe [67, S. 1168 (Tab. 3)]
const d_c1 = 0.02
# Wärmeleitfähigkeit der 1. Schicht der Decke [λ_c1] = W/(m*K)
# siehe [67, S. 1168 (Tab. 3)]
const λ_c1 = 0.72
# Dichte der 1. Schicht der Decke [ρ_c1] = kg/m3
# siehe [10]
const ρ_c1 = 1_600.0
# spezifische Wärmekapazität der 1. Schicht der Decke [c_c1] = J/(kg*K)
# siehe [10]
const c_c1 = 1_100.0
# Dicke der 2. Schicht der Decke [d_c2] = m
# siehe [67, S. 1168 (Tab. 3)]
const d_c2 = 0.075
# Wärmeleitfähigkeit der 2. Schicht der Decke [λ_c2] = W/(m*K)
# siehe [67, S. 1168 (Tab. 3)]
const λ_c2 = 2.10
# Dichte der 2. Schicht der Decke [ρ_c2] = kg/m3
# siehe [143]
const ρ_c2 = 2_400.0
# spezifische Wärmekapazität der 2. Schicht der Decke [c_c2] = J/(kg*K)
# siehe [22]
const c_c2 = 1_050.0
# Dicke der 3. Schicht der Decke [d_c3] = m
# siehe [67, S. 1168 (Tab. 3)]
const d_c3 = 0.075
# Wärmeleitfähigkeit der 3. Schicht der Decke [λ_c3] = W/(m*K)
# siehe [67, S. 1168 (Tab. 3)]
const λ_c3 = 2.10
# Dicke der 4. Schicht der Decke [d_c4] = m
# siehe [67, S. 1168 (Tab. 3)]
const d_c4 = 0.05
# Wärmeleitfähigkeit der 4. Schicht der Decke [λ_c4] = W/(m*K)
# siehe [57]
const λ_c4 = 0.035
# spezifische Wärmekapazität der 4. Schicht der Decke [c_c4] = J/(kg*K)
# siehe [57]
const c_c4 = 1_000.0
# Dichte der 4. Schicht der Decke [ρ_c4] = kg/m3
# siehe [57]
const ρ_c4 = 25.0
# Wärmeübergangskoeffizient der Decke (innen) [h_w_in] = W/(m2*K)
# siehe [55, Abschn. Praxis]
const h_c_in = 10.0
# Wärmeübergangskoeffizient der Decke (außen) [h_w_out] = W/(m2*K)

```

```

# siehe [55, Abschn. Praxis]
const h_c_out = 25.0

# BODEN:
#
# Dicke der 1. Schicht des Bodens [d_f1] = m
# siehe [67, S. 1168 (Tab. 3)]
const d_f1 = 0.03
# Wärmeleitfähigkeit der 1. Schicht des Bodens [λ_f1] = W/(m*K)
# siehe [57]
const λ_f1 = 1.10
# spezifische Wärmekapazität der 1. Schicht des Bodens [c_f1] = J/(kg*K)
# siehe [57]
const c_f1 = 837.0
# Dichte der 1. Schicht des Bodens [ρ_f1] = kg/m³
# siehe [57]
const ρ_f1 = 2_100.0
# Dicke der 2. Schicht des Bodens [d_f2] = m
# siehe [67, S. 1168 (Tab. 3)]
const d_f2 = 0.06
# Wärmeleitfähigkeit der 2. Schicht des Bodens [λ_f2] = W/(m*K)
# siehe [67, S. 1168 (Tab. 3)]
const λ_f2 = 1.40
# spezifische Wärmekapazität der 2. Schicht des Bodens [c_f2] = J/(kg*K)
# siehe [130]
const c_f2 = 1_548.0
# Dichte der 2. Schicht des Bodens [ρ_f2] = kg/m³
# siehe [42]
const ρ_f2 = 350.0
# Dicke der 3. Schicht des Bodens [d_f3] = m
# siehe [67, S. 1168 (Tab. 3)]
const d_f3 = 0.05
# Wärmeleitfähigkeit der 3. Schicht des Bodens [λ_f3] = W/(m*K)
# siehe [67, S. 1168 (Tab. 3)]
const λ_f3 = 2.10
# Dicke der 4. Schicht des Bodens [d_f4] = m
# siehe [67, S. 1168 (Tab. 3)]
const d_f4 = 0.05
# Wärmeleitfähigkeit der 4. Schicht des Bodens [λ_f4] = W/(m*K)
# siehe [67, S. 1168 (Tab. 3)]
const λ_f4 = 2.10
# Dicke der 5. Schicht des Bodens [d_f5] = m
# siehe [67, S. 1168 (Tab. 3)]
const d_f5 = 0.10
# Wärmeleitfähigkeit der 5. Schicht des Bodens [λ_f5] = W/(m*K)
# siehe [67, S. 1168 (Tab. 3)]
const λ_f5 = 0.042
# Dicke der 6. Schicht des Bodens [d_f6] = m
# siehe [67, S. 1168 (Tab. 3)]
const d_f6 = 0.15
# Wärmeleitfähigkeit der 6. Schicht des Bodens [λ_f6] = W/(m*K)
# siehe [67, S. 1168 (Tab. 3)]
const λ_f6 = 1.74
# Wärmeübergangskoeffizient der Decke (innen) [h_w_in] = W/(m²*K)
# siehe [55, Abschn. Praxis]
const h_f_in = 5.9

# WÄNDE:
#

```

```

# Dicke der 1. Schicht der Wand [d_w1] = m
# siehe [67, S. 1168 (Tab. 3)]
const d_w1 = 0.02
# Wärmeleitfähigkeit der 1. Schicht der Wand [λ_w1] = W/(m*K)
# siehe [12]
const λ_w1 = 0.80
# spezifische Wärmekapazität der 1. Schicht der Wand [c_w1] = J/(kg*K)
# siehe [12]
const c_w1 = 1_130.0
# Dichte der 1. Schicht der Wand [ρ_w1] = kg/m³
# siehe [12]
const ρ_w1 = 1_800.0
# Dicke der 2. Schicht der Wand [d_w2] = m
# siehe [67, S. 1168 (Tab. 3)]
const d_w2 = 0.135
# Wärmeleitfähigkeit der 2. Schicht der Wand [λ_w2] = W/(m*K)
# siehe [57]
const λ_w2 = 0.519
# spezifische Wärmekapazität der 2. Schicht der Wand [c_w2] = J/(kg*K)
# siehe [57]
const c_w2 = 837.0
# Dichte der 2. Schicht der Wand [ρ_w2] = kg/m³
# siehe [57]
const ρ_w2 = 1_218.0
# Dicke der 3. Schicht der Wand [d_w3] = m
# siehe [67, S. 1171 (Abschn. 3.2)]
const d_w3 = 0.05
# Wärmeleitfähigkeit der 3. Schicht der Wand [λ_w3] = W/(m*K)
# siehe [67, S. 1168 (Tab. 3)]
const λ_w3 = 0.034
# Dicke der 4. Schicht der Wand [d_w4] = m
# siehe [67, S. 1168 (Tab. 3)]
const d_w4 = 0.085
# Wärmeleitfähigkeit der 4. Schicht der Wand [λ_w4] = W/(m*K)
# siehe [57]
const λ_w4 = 0.519
# Dicke der 5. Schicht der Wand [d_w5] = m
# siehe [67, S. 1168 (Tab. 3)]
const d_w5 = 0.02
# Wärmeleitfähigkeit der 5. Schicht der Wand [λ_w5] = W/(m*K)
# siehe [11]
const λ_w5 = 0.80
# Wärmeübergangskoeffizient der Wand (innen) [h_w_in] = W/(m²*K)
# siehe [67, S. 1168 (Tab. 3)]
const h_w_in = 8.3
# Wärmeübergangskoeffizient der Wand (außen) [h_w_out] = W/(m²*K)
# siehe [67, S. 1168 (Tab. 3)]
const h_w_out = 34.0
# Effektive Dicke für Wärmespeicherung [d_eff] = m
const d_eff = 0.10

# FENSTER:
#
# Wärmedurchgangskoeffizient der Fenster [U_win] = W/(m²*K)
# siehe [67, S. 1168 (Tab. 3)]
const U_win = 3.46

# RAUMLUFT:
#

```

```

# spezifische Wärmekapazität der Raumluft [c_air] = J/(kg*K)
const c_air = 1_005.0
# Dichte der Raumluft [rho_air] = kg/m³
const rho_air = 1.2041

const U_c = 1 / (1/h_c_in + d_c1/lambda_c1 + d_c2/lambda_c2 +
                d_c3/lambda_c3 + d_c4/lambda_c4 + 1/h_c_out)
const U_c_int = 1 / (1/h_c_in + d_c1/lambda_c1 + d_c2/lambda_c2)
const U_c_ext = 1 / (d_c3/lambda_c3 + d_c4/lambda_c4 + 1/h_c_out)

const U_f = 1 / (1/h_f_in + d_f1/lambda_f1 + d_f2/lambda_f2 + d_f3/lambda_f3 +
                d_f4/lambda_f4 + d_f5/lambda_f5 + d_f6/lambda_f6)
const U_f_int = 1 / (1/h_f_in + d_f1/lambda_f1 + d_f2/lambda_f2 + d_f3/lambda_f3)
const U_f_ext = 1 / (d_f4/lambda_f4 + d_f5/lambda_f5 + d_f6/lambda_f6)

const U_fc = U_f + U_c
const U_fc_int = U_f_int + U_c_int
const U_fc_ext = (U_f + U_c) * U_fc_int / (U_fc_int - (U_f + U_c))
#const U_fc2 = (U_fc_int * U_fc_ext) / (U_fc_int + U_fc_ext)

const U_wall = 1 / (1/h_w_in + d_w1/lambda_w1 + d_w2/lambda_w2 + d_w3/lambda_w3 +
                  d_w4/lambda_w4 + d_w5/lambda_w5 + 1/h_w_out)
const U_wall_int = 1 / (1/h_w_in + d_w1/lambda_w1 + d_w2/lambda_w2)
const U_wall_ext = 1 / (d_w3/lambda_w3 + d_w4/lambda_w4 + d_w5/lambda_w5 + 1/h_w_out)

end

```

List. D.3: Julia-Quelltext von DAEConstantsModule.jl

```

module DAEConstantsModule

using JLD2, FileIO
using Interpolations

using BuildingConstantsModule
using MediumConstantsModule
using HeaterConstantsModule

# Störgrößen
export T_out, T_user, Q_add
# Parameter der differentiellen Gleichungen
export v_wall_int, v_wall_ext, v_fc_int, v_fc_ext, v_win, v_w
export C_room, C_wall, C_fc, C_htr
export v_htr, n_htr
# Parameter der algebraischen Gleichungen
export m_hc, T_hc0, c_vc3, c_vc2, c_vc1, c_vc0, alpha_min, alpha_max
export g_hc_spl, g_vc_spl #, H_pipes, H_valve

```

```

#=
#include_dependency( "../jld2s/Data__H_pipes.jld2")
#include_dependency( "../jld2s/Data__H_valve.jld2")
include_dependency( "../jld2s/Data__g_hc.jld2")
include_dependency( "../jld2s/Data__g_vc.jld2")
include_dependency( "../jld2s/Data__T_out.jld2")
include_dependency( "../jld2s/Data__T_user.jld2")
#=#

# Parameter der algebraischen Gleichungen +
# Störgrößen
include( "../FuncsModule/Qdot_add.jl")
#@load "../jld2s/Data__H_pipes.jld2" H_pipes
#@load "../jld2s/Data__H_valve.jld2" H_valve
@load "../jld2s/Data__g_hc.jld2" g_hc_spl m_hc T_hc0
@load "../jld2s/Data__g_vc.jld2" g_vc_spl c_vc3 c_vc2 c_vc1 c_vc0 a_min a_max
@load "../jld2s/Data__T_out.jld2" T_out
@load "../jld2s/Data__T_user.jld2" T_user

# WillisRealHut_N2:
# Parameter der differentiellen Gleichungen

# [v_wall_int] = J/(s*K) = kg*m²/(s³*K)
const v_wall_int = A_wall * U_wall_int
# [v_wall_ext] = J/(s*K) = kg*m²/(s³*K)
const v_wall_ext = A_wall * U_wall_ext
# [v_fc_int] = J/(s*K) = kg*m²/(s³*K)
const v_fc_int = A_fc * U_fc_int
# [v_fc_ext] = J/(s*K) = kg*m²/(s³*K)
const v_fc_ext = A_fc * U_fc_ext
# [v_win] = J/(s*K) = kg*m²/(s³*K)
const v_win = A_win * U_win
# [v_w] = J/(m³*K) = kg/(m*s²*K)
const v_w = c_w * rho_w
# [C_room] = J/K = kg*m²/(s²*K)
const C_room = c_air * rho_air * H_room * L_room * B_room
# [C_wall] = J/K = kg*m²/(s²*K)
const C_wall = c_w1 * rho_w1 * d_w1 * A_wall +
              c_w2 * rho_w2 * (d_eff - d_w1) * A_wall
# [C_fc] = J/K = kg*m²/(s²*K)
const C_fc = c_c1 * rho_c1 * d_c1 * A_ceiling + c_c2 * rho_c2 * d_c2 * A_ceiling +
            c_f1 * rho_f1 * d_f1 * A_floor + c_f2 * rho_f2 * d_f2 * A_floor
# [C_htr] = J/K = kg*m²/(s²*K)
const C_htr = flV * rho_w * c_w / nEle

end

```

D.3 DAE-System

List. D.4: Julia-Quelltext von f3.jl

```
function f3( t, y, z, d, u, θ_d)

    res = similar( y)

    # Differentielle Zustandsvariablen
    T_room, T_wall, T_fc, T_htr, T_rtn = y
    # Algebraische Zustandsvariablen
    T_flw, α, Ḃ = z

    # Störgrößen
    T_out, T_user, Q_add = d

    # Steuergröße
    n_pump = u(t)

    # Parameter
    n_htr, C_room, C_wall, C_fc, C_htr,
    v_wall_int, v_wall_ext, v_fc_int, v_fc_ext, v_win, v_htr, v_w = θ_d

    # Gleichung für T_room
    res[1] = 1/C_room * ( v_wall_int * (T_wall - T_room) +
        v_fc_int * (T_fc - T_room) + v_win * (T_out(t) - T_room) +
        v_htr * hpow(T_htr - T_room, n_htr) +
        v_htr * hpow(T_rtn - T_room, n_htr) + Q_add(t))

    # Gleichung für T_wall
    res[2] = 1/C_wall * ( v_wall_int * (T_room - T_wall) +
        v_wall_ext * ( T_out(t) - T_wall))

    # Gleichung für T_fc
    res[3] = 1/C_fc * ( v_fc_int * (T_room - T_fc) +
        v_fc_ext * ( T_out(t) - T_fc))

    # Gleichung für T_htr
    res[4] = 1/C_htr * (-v_htr * hpow(T_htr - T_room, n_htr) +
        v_w * Ḃ * (T_flw - T_htr))

    # Gleichung für T_rtn
    res[5] = 1/C_htr * (-v_htr * hpow(T_rtn - T_room, n_htr) +
        v_w * Ḃ * (T_htr - T_rtn))

    return res

end
```

List. D.5: Julia-Quelltext von g3.jl

```
function g3( t, y, z, d, u,  $\theta_a$ )

    res = similar( z )

    # Differentielle Zustandsvariablen
    T_room, T_wall, T_fc, T_htr, T_rtn = y
    # Algebraische Zustandsvariablen
    T_flw,  $\alpha$ ,  $\dot{V}$  = z

    # Störgrößen
    T_out, T_user,  $\dot{Q}_{add}$  = d

    # Steuergröße
    n_pump = u(t)

    # Parameter
     $\theta_{hc}$ ,  $\theta_{vc}$  =  $\theta_a$ 

    # Gleichung für T_flw
    res[1] = T_flw - g_hc( T_out(t),  $\theta_{hc}$ )

    # Gleichung für  $\alpha$ 
    res[2] =  $\alpha$  - g_vc( T_room - T_user(t),  $\theta_{vc}$ )

    # Gleichung für  $\dot{V}$ 
    res[3] =  $\dot{V}$  - g_opp( n_pump,  $\alpha$ )

    return res

end
```

List. D.6: Julia-Quelltext von rhsHS3.jl

```
function rhsHS3!( out,  $\dot{x}$ , x, du $\theta$ , t)

    # Zustandsvariablen
     $y$ ,  $z$  = x[1:5], x[6:8]
     $\dot{y}$ ,  $\dot{z}$  =  $\dot{x}$ [1:5],  $\dot{x}$ [6:8]

    # Störgrößen
     $d$  = du $\theta$ [1]
    # Steuergröße
     $u$  = du $\theta$ [2]
    # Parameter
     $\theta$  = du $\theta$ [3]
     $\theta_d$ ,  $\theta_a$  =  $\theta$ 

    # Gleichungen der rechten Seite
    out[1:5] =  $\dot{y}$  - f3( t,  $y$ ,  $z$ ,  $d$ ,  $u$ ,  $\theta_d$ )
    out[6:8] = g3( t,  $y$ ,  $z$ ,  $d$ ,  $u$ ,  $\theta_a$ )

end
```

D.4 ODE-System

List. D.7: Julia-Quelltext von f1.jl

```
function f1( x, d, u, θ, t)

    res = similar( x)

    # Störgrößen
    T_out, T_user, Q_add = d

    # Steuergröße
    n_pump = u(t)

    # parameters:
    θ_d, θ_a = θ
    n_htr, C_room, C_wall, C_fc, C_htr,
    v_wall_int, v_wall_ext, v_fc_int, v_fc_ext, v_win, v_htr, v_w = θ_d
    θ_hc, θ_vc = θ_a

    # Differentielle Zustandsvariablen
    T_room, T_wall, T_fc, T_htr, T_rtn = x
    # Algebraische Zustandsvariablen
    T_flw = g_hc( T_out(t), θ_hc)
    α = g_vc( T_room - T_user(t), θ_vc)
    V = g_opp( n_pump, α)

    # Gleichung für T_room
    res[1] = 1/C_room * ( v_wall_int * (T_wall - T_room) +
        v_fc_int * (T_fc - T_room) + v_win * (T_out(t) - T_room) +
        v_htr * hpow(T_htr - T_room, n_htr) +
        v_htr * hpow(T_rtn - T_room, n_htr) + Q_add(t))

    # Gleichung für T_wall
    res[2] = 1/C_wall * ( v_wall_int * (T_room - T_wall) +
        v_wall_ext * ( T_out(t) - T_wall))

    # Gleichung für T_fc
    res[3] = 1/C_fc * ( v_fc_int * (T_room - T_fc) +
        v_fc_ext * ( T_out(t) - T_fc))

    # Gleichung für T_htr
    res[4] = 1/C_htr * (-v_htr * hpow(T_htr - T_room, n_htr) +
        v_w * V * (T_flw - T_htr))

    # Gleichung für T_rtn
    res[5] = 1/C_htr * (-v_htr * hpow(T_rtn - T_room, n_htr) +
        v_w * V * (T_htr - T_rtn))

    return res

end
```

List. D.8: Julia-Quelltext von rhsHS1.jl

```
function rhsHS1!( $\dot{x}$ ,  $x$ ,  $du\theta$ ,  $t$ )  
  
    # Störgrößen  
     $d = du\theta[1]$   
    # Steuergröße  
     $u = du\theta[2]$   
    # Parameter  
     $\theta = du\theta[3]$   
  
    # Gleichungen der rechten Seite  
     $\dot{x}[1:5] = f1(x, d, u, \theta, t)$   
  
end
```

D.5 Hilfsfunktion für die Initialisierung der KI-Strategie

List. D.9: Julia-Quelltext von q-hat_star.jl

```
hatR̄ = R_max

function q̂_star( AS )

    A, S = AS[1], AS[2]
    res_q̂ = 0.0

    n_pump = n_min + 1/2 * (S[1] + 1) * (n_max - n_min)
    Δ̄ = Δn_min + 1/2 * (A + 1) * (Δn_max - Δn_min)

    n_pump' = n_pump + Δ̄
    S' = 2 * (n_pump' - n_min) / (n_max - n_min) - 1
    R' = 1.0 - abs( S' )

    res_q̂ = R' - hatR̄

    n_mid = 1/2 * (n_min + n_max)

    while n_pump' > n_mid
        n_pump' = n_pump' + max( Δn_min, n_mid - n_pump' )
        S' = 2 * (n_pump' - n_min) / (n_max - n_min) - 1
        R' = 1.0 - abs( S' )

        res_q̂ = res_q̂ + R' - hatR̄
    end

    while n_pump' < n_mid
        n_pump' = n_pump' + min( Δn_max, n_mid - n_pump' )
        S' = 2 * (n_pump' - n_min) / (n_max - n_min) - 1
        R' = 1.0 - abs( S' )

        res_q̂ = res_q̂ + R' - hatR̄
    end

    return res_q̂

end
```