

**Beiträge zur aktiven Strömungsbeeinflussung:
Systemmodellierung mit Methoden des maschinellen Lernens
und ganzzahlig beschränkte Regelung für zyklische Prozesse**

vorgelegt von

M. Sc.

Florian Robert Eduard Arnold

ORCID: 0000-0001-6413-0443

von der Fakultät III – Prozesswissenschaften

der Technischen Universität Berlin

zur Erlangung des akademischen Grades

Doktor der Ingenieurwissenschaften

– Dr.-Ing. –

genehmigte Dissertation

Promotionsausschuss:

Vorsitzende: Prof. Dr. Steffi Knorn

Gutachter: Prof. Dr.-Ing. habil. Rudibert King

Gutachter: Prof. Charles Robert Koch, PhD, PEng

Tag der wissenschaftlichen Aussprache: 18. März 2022

Berlin 2022

Kurzfassung

Die aktive Strömungsbeeinflussung kann in vielen technischen Bereichen zur Optimierung von aerodynamischen Eigenschaften bzw. zur Kompensation von Störungen eingesetzt werden. Allerdings ist die technische Realisierung der aktiven Strömungsbeeinflussung in vielen Fällen sehr herausfordernd. Dies gilt insbesondere bei der Umsetzung in einem geschlossenen Regelkreis. Zwei dieser Herausforderungen, auf die in dieser Arbeit eingegangen wird, sind die Beschreibung der Strömung in einem recheneffizienten Modell sowie der Umgang mit mechanischen Beschränkungen der Aktuation. Die beschriebenen Methoden werden im Rahmen dieser Arbeit genutzt, um ausgewählte strömungsmechanische Herausforderungen beim Betrieb einer Gasturbine mit Gleichraumverbrennung zu adressieren.

Im ersten Teil der Arbeit wird eine Methode zur Zustandsraummodellierung auf Basis sogenannter *Physics-Informed Neural Networks* vorgestellt. Dieser Ansatz erlaubt es, physikalische Informationen aus einer Systembeschreibung mit partiellen Differentialgleichungen in der Anpassung eines Zustandsraummodells zu berücksichtigen. Damit können auch für Systemzustände, die nicht explizit in der Anpassung des Modells berücksichtigt wurden, zuverlässige Ergebnisse erzielt werden. Das vorgestellte Modell kann allerdings im Gegensatz zu einer numerischen Lösung der partiellen Differentialgleichungen in Echtzeit ausgewertet werden und ermöglicht so, die Anwendung in einem modellgestützten Mess- oder Regelungsverfahren. Im Rahmen dieser Arbeit wird die Modellierungsmethode jeweils für ein räumlich eindimensionales System auf Basis der Burgers-Gleichung bzw. den Euler-Gleichungen getestet. Es wird sowohl die Verwendung in einer Zustandsschätzung mit einem erweiterten Kalman-Filter untersucht als auch der Einsatz in einer modellprädiktiven Regelung.

Der zweite Teil der Arbeit befasst sich mit der Regelung von zyklischen Prozessen, in denen die Aktuation beschränkt ist. Im Speziellen wird auf die Beschränkung auf einen ganzzahligen Wertebereich eingegangen. Dieser Fall ist in der aktiven Strömungsbeeinflussung vor allem bei der Verwendung von schaltenden Bauteilen mit einer beschränkten Aktuationsfrequenz relevant. Es wird ein Syntheseverfahren für eine normoptimale iterativ lernende Regelung (ILR) mit einem ganzzahlig beschränkten Eingang vorgestellt. Dabei stellt das Syntheseverfahren sicher, dass die relevanten Eigenschaften der reellwertigen normoptimalen ILR erhalten bleiben. Zusätzlich wird der Sonderfall einer binären ILR untersucht und ein Verfahren vorgestellt, mit dem unter bestimmten Umständen der Rechenaufwand der Regelung deutlich reduziert werden kann. Die Methode wird an einem akustischen Versuchsaufbau getestet, der ein Plenum hinter den Verbrennungsgrohren eine Gasturbine mit Gleichraumverbrennung nachbildet.

Zusätzlich wird als Alternative zur normoptimalen ILR ein modellfreier Ansatz auf Basis einer Extremwertregelung (EWR) vorgestellt. Auch in diesem Verfahren können ganzzahlige Eingangsbeschränkungen für zyklische Prozesse berücksichtigt werden. Diese zyklische EWR wird an einem experimentellen Aufbau, einer sogenannten stoßfreien Explosionsverbrennung (SEC), getestet. Ziel für die SEC ist dabei, den Druckanstieg zu maximieren und damit möglichst gut eine Gleichraumverbrennung anzunähern.

Abstract

Active flow control is a widely used technique to optimize aerodynamic features or to compensate for disturbances. However, implementing active flow control for a specific application can often be challenging, especially if a closed-loop control is required. This thesis addresses two of these challenges. Besides a method to define a computationally efficient model, this contribution introduces various methods for handling input constraints caused by mechanical restrictions. The described control approaches are used to cope with the flow-related challenges of operating a gas turbine with constant volume combustion.

The first part of the thesis presents a method for state-space modeling based on so-called physics-informed neural networks. This approach allows fitting a state-space model considering physical information from a system description with partial differential equations. By that, a system description can be achieved that will give reliable results even for states not considered for the model fit. As a numerical solution of the partial differential equations is not required, the fitted model can be evaluated in real-time. Thus, it is suitable for model-based measurement and control applications. Within the scope of this work, the modeling method is tested for a spatially one-dimensional system based on the Burgers equation and the Euler equations, respectively. Both models are used to implement state estimation based on an extended Kalman filter and a model predictive control to illustrate the performance of the state-space model.

The second part of the thesis deals with the control of cyclic processes with a constrained input. In particular, the restriction to an integer-valued domain is discussed. This case is especially relevant in active flow control when switching components with a limited frequency are used to actuate the flow. An approach for an integer-valued norm-optimal iterative learning control (ILC) is presented and tested. This control synthesis ensures that the relevant properties of the real-valued norm-optimal ILC are preserved. Additionally, the particular case of a binary ILC is investigated, and a method to significantly reduce the computational effort of the control for a specific set of control tasks is introduced. The method is applied to an acoustic test setup, which mimics a plenum behind the combustion tubes of a gas turbine operated with constant volume combustion.

In addition, a model-free approach based on extremum seeking control (ESC) is presented to provide an alternative to norm-optimal ILC. Similar to the presented ILC approach, the introduced cyclic ESC can also cope with integer-valued input constraints. The control is tested on an experimental setup of a so-called shockless explosion combustion (SEC). In this setup, the cyclic ESC aims to maximize the pressure rise from the combustion to approximate constant volume combustion in an optimal manner.

Inhaltsverzeichnis

1	Einleitung	1
1.1	Konzept einer Gasturbine mit Gleichraumverbrennung	1
1.2	Motivation	4
1.2.1	Modellierung	4
1.2.2	Regelung	5
1.3	Aufbau der Arbeit	6
2	Stand der Forschung	9
2.1	Strömungsmodellierung	9
2.2	Regelung mit beschränkter Stellgröße	12
2.2.1	Ganzzahlige quadratische Optimierung	13
2.3	Aktive Strömungsbeeinflussung	14
3	Modellierung	17
3.1	Konventionelle Modellierungsansätze für Strömungen	17
3.1.1	Navier-Stokes-Gleichungen	17
3.1.2	Euler-Gleichungen	19
3.1.3	Burgers-Gleichung	19
3.1.4	Numerische Lösung	20
3.2	Neuronale Netze	23
3.2.1	Netzarchitektur	23
3.2.2	Training	25
3.2.3	Zeitreihenprädiktion	27
3.3	Physics-Informed Neural Networks	28
3.4	Zustandsraummodellierung auf Basis von Physics-Informed-Neural-Networks	32
3.4.1	PINN-Approximation der Burgers-Gleichung	36
3.4.2	PINN-Approximation der Euler-Gleichungen	37
4	Zustandsschätzung	49
4.1	Erweitertes Kalman-Filter	49
4.2	Erweitertes Kalman-Filter für die Burgers-Gleichung	51
4.3	Erweitertes Kalman-Filter für die Euler-Gleichungen	54
5	Modellprädiktive Regelung	61
5.1	Regelung Burgers-Gleichung	62
5.2	Regelung Euler-Gleichungen	64
6	Iterativ lernende Regelung	67
6.1	Grundlagen der iterativ lernenden Regelung	67
6.2	Normoptimale iterativ lernende Regelung für lineare Systeme	69
6.2.1	Beschränkungen	71
6.2.2	Stabilität und Konvergenz	72

6.3	Synthese einer ganzzahligen iterativ lernenden Regelung	76
6.3.1	Ad-hoc-Ansätze	77
6.3.2	Optimale-Menge-Synthese	81
6.3.3	Simulationsstudie für ein generisches Beispiel	85
6.4	Binäre iterativ lernende Regelung	89
6.4.1	Vergleich mit Rundungsansatz	89
6.5	Anwendung Feuersynchronisation	93
6.5.1	Akustischer Plenumsprüfstand	93
6.5.2	Reduzierung des Rechenaufwands	96
6.5.3	Ergebnisse am akustischen Plenumsversuch	98
6.5.4	Wahl der Lösungsmenge	102
7	Extremwertregelung	107
7.1	Extremwertregelung für zyklische Prozesse	107
7.1.1	Extremwertsuche	108
7.1.2	Beschränkungen	110
7.2	Ergebnisse für 1D-Euler-Simulation	110
7.3	Extremwertregelung an einem SEC-Experiment	114
7.3.1	Experimenteller Aufbau	115
7.3.2	Ergebnisse	116
8	Fazit	121
8.1	Ausblick	122
9	Anhang	125
A	Finite-Volumen-Methode	125
B	BFGS-Algorithmus	127
C	Branch-and-Bound-Suche für ganzzahlige quadratische Programme	130

Nomenklatur

Große lateinische Buchstaben

A	Dynamikmatrix in der Zustandsgleichung eines Zustandsraummodells
<i>A</i>	Amplitude eines Signals
B	Eingangsmatrix in der Zustandsgleichung eines Zustandsraummodells
<i>B</i>	Menge der Bias-Parameter eines neuronalen Netzes
C	Dynamikmatrix in der Messgleichung eines Zustandsraummodells
D	Eingangsmatrix in der Messgleichung eines Zustandsraummodells
<i>D</i>	eine Diagonalmatrix
<i>D</i>	Dämpfungskonstante
<i>E</i>	spezifische Energie
\mathcal{E}	Erwartungswert
F	Jacobi-Matrix der Zustandsgleichung
F_S	Normal- bzw. Scherspannung
F_V	Volumenkraft
\mathcal{F}	Menge an neuronalen Netzen
G	Modellmatrix eines linearen Eingangs-/Ausgangsmodells
<i>G</i>	Übertragungsfunktion
H	Hesse-Matrix
H_2	molekularer Wasserstoff
I	vektorielle Darstellung einer Impulsantwort
\dot{I}_{in}	eintretender Impulsstrom
\dot{I}_{out}	austretender Impulsstrom
\dot{I}_K	Impulsänderung in einem Kontinuum
<i>J</i>	Kosten
J^{opt}	global optimale Kosten
J^+	obere Schranke für die Kosten
J^*	unbekannte Kosten
J'	bekannte Kosten
K	Korrekturmatrix des Kalman-Filter
L	Anpassungsfilter in einer ILR
<i>L</i>	Trainingsfunktion
\mathcal{M}	Menge der diskreten Aktuationsschritte
N_2	molekularer Stickstoff
$N_{i,j}$	<i>i</i> -tes Neuron auf der <i>j</i> -ten Schicht eines neuronalen Netzes
<i>N</i>	Zyklus, in dem eine konvergierte Stellfolge erreicht wird
\mathcal{N}	Differentialoperator
O_2	molekularer Sauerstoff
P_k	Kovarianzmatrix des Schätzfehlers im Zeitschritt <i>k</i>
Q	Anpassungsfilter in einer ILR/quadratischer Parameter eines QPs
\bar{Q}	quadratischer Parameter eines Ersatz QPs
Q_k	Kovarianzmatrix des Systemrauschens im Zeitschritt <i>k</i>
R_k	Kovarianzmatrix des Messrauschens im Zeitschritt <i>k</i>

Inhaltsverzeichnis

R	spezifische Gaskonstante
\mathbf{S}	Diagonalmatrix der Singulärwerte
\mathcal{S}	Menge zulässiger Lösungen
\mathbf{T}	Transformationsmatrix
T	Zeitspanne
T_s	Schaltzeit
T_p	Dauer einer Periode
\mathbf{U}	Eingang in eine LSTM-Einheit
\mathcal{U}	ganzahliger Unterraum
\mathcal{U}^T	transformierter, ganzzahliger Unterraum
\mathcal{V}	Menge an zulässigen Schritten
\mathcal{V}^g	generische Menge an zulässigen Schritten
\mathbf{V}	Matrix der Eigenvektoren
\mathbf{W}	Tensor der Parameter einer Schicht in einem neuronalen Netz
\mathbf{W}_E	Gewichtungsmatrix des Fehlers
\mathbf{W}_U	Gewichtungsmatrix der Stellfolge
$\mathbf{W}_{\Delta U}$	Gewichtungsmatrix der Änderung der Stellfolge
\mathcal{W}	Menge der Parameter eines neuronalen Netzes
\mathbf{X}	Eingang in ein neuronales Netz
\mathbf{X}_T	Eingangstensor aus der Trainingsmenge eines neuronalen Netzes
\mathcal{X}	Menge von Eingangsdaten in ein neuronales Netz
\mathcal{X}_T	Menge der Eingangstensoren für das Training eines neuronalen Netzes
\mathbf{Y}	Ausgang aus einem neuronalen Netz
\mathbf{Y}_T	Ausgangstensor aus der Trainingsmenge eines neuronalen Netzes
\mathbf{Y}_P	Ausgangstensor, der mit einem neuronalen Netz prädiziert wurde
\mathcal{Y}	Menge der Eingangsdaten in ein neuronales Netz
\mathcal{Y}_T	Menge der Ausgangstensoren für das Training eines neuronalen Netzes
\mathcal{Y}_P	Menge der Ausgangstensoren, die mit einem neuronalen Netz prädiziert wurden
\mathbf{Z}	Linearisierungsmatrix für Messgleichung

Kleine lateinische Buchstaben

a	Parameter für die Größe des Lösungsraum
\mathbf{b}	Tensor der Bias-Parameter einer Schicht in einem neuronalen Netz
b	Versatz in Modell
b	skalarer Bias-Parameter in einem Neuron
b	Versatz der Referenzfunktion
\mathbf{c}	Zellzustand
\mathbf{c}	Vektor in linearer Ungleichung
\mathbf{c}_{lb}	untere Schranke in linearer Ungleichung
\mathbf{c}_{ub}	obere Schranke in linearer Ungleichung
c_{lb}	skalare untere Schranke
c_{ub}	skalare obere Schranke
\mathbf{d}	vektorielle Störung
\mathbf{d}	Spalte einer Diagonalmatrix
d	Ähnlichkeitsmaß
\mathbf{e}	vektorieller Regelfehler
e	skalarer Regelfehler
e_L	Güte des Trainings eines neuronalen Netzes
\mathbf{f}	gerichtete Kraft/Kraftvektor

\mathbf{f}	Zustandsfunktion/Modellfunktion
\mathbf{f}^*	unbekannte Funktion
f_d	Funktion zur Bestimmung einer Kennzahl für die Ähnlichkeit zweier Vektoren
f_f	Feuerfrequenz
f_h	Abbildung in einer partiellen Differentialgleichung
f_r	Frequenz der Referenz
f_u	Abtastfrequenz der Aktuation
f_y	Abtastfrequenz der Messung
\mathbf{g}	Differenz der Gradienten zwischen zwei Iterationen des BFGS-Algorithmus
\mathbf{g}	Stellgesetz
\mathbf{h}	Ersatzvariable im BFGS-Algorithmus
\mathbf{h}	Lösung einer vektoriellen partiellen Differentialgleichung
h	Lösung einer skalaren partiellen Differentialgleichung
h_0	Anfangsbedingung einer skalaren partiellen Differentialgleichung
i	Zählvariable
i_0	Zählvariable für Anfangsbedingungen
i_h	Zählvariable für Trainingsdaten der Randbedingung
i_θ	Zählvariable für Trainingsdaten des Raum-Zeit-Gitters
j	Zählvariable
k	Zyklusvariable
l	Zeitschritt
l_c	Zeitschritt für Definition der Kovarianz
m_Ψ	Anzahl der Schichten eines neuronalen Netzes
\dot{m}	Massenstrom
n	Stoffmenge
n	Dimension des Systemeingangs und des Systemausgangs bei gleicher Größe
n_C	Anzahl an Verbrennungsrohren
n_0	Anzahl verschiedener Anfangsbedingungen
n_1	Anzahl der Einsen in einem Vektor
n_L	Anzahl der Lautsprecher
n_M	Anzahl der Messpunkte
$n_{\mathcal{M}}$	Anzahl der diskreten Aktuationsschritte
n_N	Anzahl der Punkte der numerischen Lösung
$n_{\mathcal{S}}$	Dimension der Lösungsmenge \mathcal{S}
n_T	Anzahl der Trainingsdaten
n_U	Dimension des Stellhorizonts
n_u	Dimension der Stellfolge
n_Y	Dimension des Prädiktionshorizonts
n_y	Dimension des Systemausgangs
n_x	Dimension des Zustands
n_Z	Anzahl der diskreten Zellen in einer Finite-Volumen-Methode (FVM)
n_Ψ	Anzahl der Neuronen in einer Schicht eines neuronalen Netzes
n_ω	Anzahl der Stützstellen
\mathbf{p}	Binärvektor
p	Eintrag des Binärvektors
p	Druck
\bar{p}	Optimierungsschritt
p_a	Anpassungsfaktor
\mathbf{q}	linearer Parameter eines QPs

Inhaltsverzeichnis

\bar{q}	linearer Parameter eines Ersatz QPs
q	konstantes Parameter eines QPs
\mathbf{r}	vektorielle Referenz
r	skalare Referenz
\mathbf{s}	Schritt zwischen zwei Stellfolgen
s	Laplace-Variable
t	Zeit
t_I	Einspritzzeit
\mathbf{u}	ganzzahlige Stellfolge
$\tilde{\mathbf{u}}$	Iterationsvariable für ganzzahlige Stellfolge
$\bar{\mathbf{u}}$	ganzzahlige Stellfolge eines Ersatzproblems
$\tilde{\tilde{\mathbf{u}}}$	Iterationsvariable für ganzzahlige Stellfolge eines Ersatzproblems
u	skalare ganzzahlige Stellgröße
\mathbf{v}	Eigenvektor
\mathbf{v}^s	skalierter Eigenvektor
\mathbf{v}	Geschwindigkeitsvektor
v	Eingang in Aktivierungsfunktion
v_x	Geschwindigkeit in x-Richtung
v_y	Geschwindigkeit in y-Richtung
v_z	Geschwindigkeit in z-Richtung
w	skalärer Parameter eines neuronalen Netzes
\mathbf{w}	Vektor der Parameter eines Neurons
$w_{\Delta U}$	skalares Gewicht für die Änderung einer Stellfolge
$w_{\Delta U, \min}$	untere Grenze des skalaren Gewichts für die Änderung einer Stellfolge
\mathbf{x}	Ortsvektor
\mathbf{x}	Zustandsvektor
$\underline{\mathbf{x}}$	Supervektordarstellung der Systemzustände
x	x-Richtung in einem kartesischen Koordinatensystem
x_M	Messposition
x	skalärer Zustand
\mathbf{y}	verborgener Zustand
\mathbf{y}	Systemausgang
y	y-Richtung in einem kartesischen Koordinatensystem
y	skalärer Systemausgang/skalärer Ausgang eines Neurons
z	Messfunktion
z	z-Richtung in einem kartesischen Koordinatensystem

Kleine griechische Buchstaben

α	Eingangsfunktion eines Neurons
β	Aktivierungsfunktion eines Neurons
β	vektorielle Aktivierungsfunktion
γ	Systemausgang eines Mehrgrößensystems
γ	Konvergenzrate
γ	Isentropenexponent
δ	Perturbation
δ_K	Kronecker-Delta
ε	Schätzfehler
ϵ	Konvergenzschranke
ζ	Schrittweite im BFGS-Algorithmus

θ	<i>Physics-Informed Neural Networks</i> (PINN)
ϑ	Temperatur
$\underline{\ell}$	Supervektordarstellung der Referenz eines Mehrgrößensystem
κ	dynamische Viskosität
λ	Eigenwert
$\boldsymbol{\mu}$	vektorielle reellwertige Stellfolge
$\tilde{\boldsymbol{\mu}}$	Iterationsvariable für reellwertige Stellfolge
μ	reellwertige skalare Stellgröße
$\boldsymbol{\nu}$	ganzzahlige Stellgröße bzw. Supervektordarstellung für Mehrgrößensysteme
$\tilde{\boldsymbol{\nu}}$	Iterationsvariable für eine ganzzahlige Stellfolge in Supervektordarstellung
ν	Eintrag in einer ganzzahligen Stellfolge in Supervektordarstellung/Viskosität
ξ	skalarer Interpolationsoperator
$\boldsymbol{\xi}$	vektorieller Interpolationsoperator
ρ	Dichte
$\boldsymbol{\sigma}$	vektorielle Sigmoidfunktion
σ	Sigmoidfunktion
σ_{Ind}	Index des maximalen Wertes in einem Vektor
σ_{max}	maximaler Singulärwert
τ	Integrationsvariable der Zeit
$\boldsymbol{\chi}$	vektorieller Eingang in ein Neuron
χ	skalarer Eingang in ein Neuron
$\boldsymbol{\omega}$	Stützstellen
ω_0	Eigenfrequenz

Große griechische Buchstaben

Δp	maximale Druckänderung
Δt	Zeitschrittweite
$\Delta \mathbf{U}$	Differenz zwischen zwei ganzzahligen Stellfolgen in Supervektordarstellung
$\Delta \mathbf{u}$	Änderung einer vektoriellen Stellfolge
$\Delta \boldsymbol{\mu}$	Änderung einer reellwertigen Stellfolge
Φ	Äquivalenzverhältnis
$\boldsymbol{\Psi}$	Modell aus Impulsantworten
Ψ	neuronales Netz
$\boldsymbol{\Omega}$	Matrix in linearer Ungleichung

Normen, Zahlenmengen, Operatoren und spezielle Vektoren und Matrizen

$\mathbf{1}$	Vektor aus Einsen
$\mathbf{0}$	Vektor bzw. Matrix aus Nullen
\mathbf{I}	Einheitsmatrix
∇	Nabla-Operator
\triangle	Laplace-Operator
∂	partielle Ableitung
$\ \cdot\ _2$	2-Norm
$\ \cdot\ _F$	Frobenius-Norm
$\ \cdot\ _\infty$	Unendlich-Norm
$\hat{\cdot}$	Schätzung
$\arg \min \cdot$	Argument, welches einen Ausdruck minimiert
$\arg \max \cdot$	Argument, welches einen Ausdruck maximiert

Inhaltsverzeichnis

\mathbb{N}	Menge der natürlichen Zahlen
\mathbb{R}	Menge der reellen Zahlen
\mathbb{B}	Menge der binären Zahlen

Abkürzungen

BFGS	Broyden–Fletcher–Goldfarb–Shanno
DGL	Differentialgleichung
EKF	erweitertes Kalman-Filter
EWK	Extremwertregelung
FVM	Finite-Volumen-Methode
ILR	iterativ lernende Regelung
LSTM	<i>Long Short-Term Memory</i>
MPC	modellprädiktive Regelung (<i>Model Predictive Control</i>)
MSE	<i>Mean Squared Error</i>
OMS	Optimale-Menge-Synthese
PDC	pulsierende Detonationsverbrennung (<i>Pulse Detonation Combustion</i>)
PINN	<i>Physics-Informed Neural Networks</i>
POD	<i>Proper Orthogonal Decomposition</i>
PWM	Pulsweitenmodulation
QP	quadratisches Programm
SEC	stoßfreie Explosionsverbrennung (<i>Shockless Explosion Combustion</i>)
SUR	<i>Sum-Up-Rounding</i>

1 Einleitung

Die gezielte Beeinflussung und Optimierung strömungsmechanischer Prozesse ist oft eine technische Herausforderung. Neben der komplexen mathematischen Beschreibung einzelner Strömungsphänomene und der hohen Sensitivität auf einzelne Stoff- und Umgebungsparameter, sind häufig auch die Aktuationsmöglichkeiten beschränkt. Aus diesem Grund ist die sogenannte aktive Strömungsbeeinflussung, bei der Bauteile wie z.B. schaltende Ventile verwendet werden, ein besonders anspruchsvoller Bereich im Bezug auf die verwendeten Steuerungs- und Regelungsansätze. Die vorliegende Arbeit untersucht und präsentiert Lösungsansätze zur Umsetzung aktiver Strömungsbeeinflussung in einem geschlossenen Regelkreis. Dabei wird sowohl auf die Schwierigkeiten in der Modellierung strömungsmechanischer Prozesse als auch auf Beschränkungen in der Aktuation eingegangen.

Auch wenn die präsentierten Ansätze für eine Vielzahl von strömungsmechanischen Anwendungen geeignet sind, wurden die Methoden mit dem Fokus auf die Implementierung in einer Gasturbine mit Gleichraumverbrennung entwickelt. Da sich die gezeigten Beispiele in dieser Arbeit auf einige Aspekte eines derartigen Systems beziehen, wird im Folgenden kurz auf das Konzept einer Gasturbine mit instationärer Gleichraumverbrennung eingegangen. Diese kurze Einführung soll die Beispiele in den einzelnen Kapiteln verständlicher machen und die Herausforderungen in der aktiven Strömungsbeeinflussung verdeutlichen. Im Anschluss daran wird die regelungstechnische Problemstellung diskutiert.

1.1 Konzept einer Gasturbine mit Gleichraumverbrennung

In den vergangenen Jahrzehnten wurde die Effizienz von Gasturbinen stetig verbessert (Badran, 1999; Walsh & Fletcher, 2004; Horlock, 2013). Durch Optimierung der einzelnen Komponenten können aktuell Kreislaufwirkungsgrade von etwas über 40% erreicht werden (Jansohn, 2013). Damit ist die Gasturbine nicht nur im Luftfahrtbereich das gängige Antriebskonzept (Ansell & Haran, 2020), sondern findet auch Verwendung bei der stationären Erzeugung von Elektrizität. Soll die Gasturbine weiterhin ein attraktives Konzept bleiben, ist es jedoch notwendig, ihre Effizienz weiter zu verbessern (Poullikkas, 2005). Untersuchungen aus den letzten Jahren zeigen allerdings, dass eine signifikante Erhöhung des Wirkungsgrads über die Optimierung der Komponenten nur schwer oder nicht zu realisieren ist. Für eine signifikante Steigerung der Effizienz ist daher eine Anpassung des grundlegenden Konzepts der Gasturbine notwendig (Stathopoulos, 2020).

Im Rahmen des DFG-Sonderforschungsbereichs 1029, der an der TU Berlin und FU Berlin angesiedelt ist, wird seit 2012 an einem Konzept für eine Gasturbine auf Basis einer Gleichraumverbrennung gemäß des Humphrey-Kreisprozesses (Stathopoulos, 2018) gearbeitet. Durch Ausnutzung des bei der Verbrennung entstehenden Druckanstiegs und der geringeren Entropieerzeugung wird eine Wirkungsgradverbesserung von bis zu zehn Prozentpunkten angestrebt (Gray et al., 2016).

Ein schematischer Aufbau einer solchen Gasturbine ist in Abb. 1.1 dargestellt. Im Zentrum

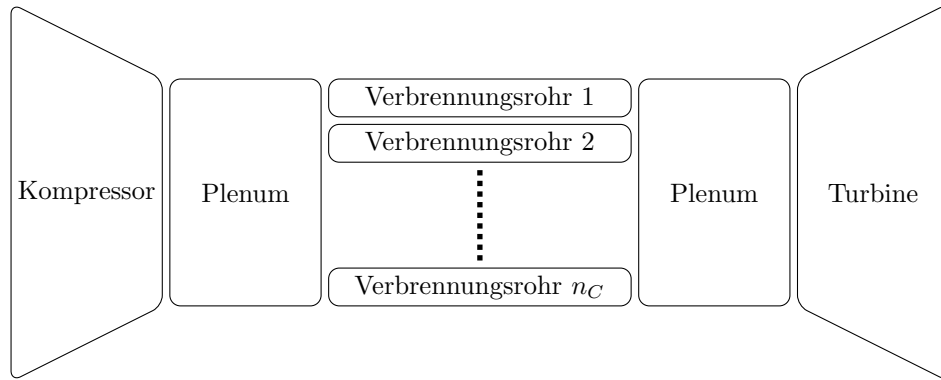


Abbildung 1.1: Schematischer Aufbau einer Gasturbine mit einer Gleichraumverbrennung

des Aufbaus steht ein Gruppe aus Verbrennungsrohren in denen eine angenäherte Gleichraumverbrennung realisiert wird. Für diesen instationären Prozess wurden im Rahmen des Sonderforschungsbereichs 1029 vor allem zwei verschiedene Verbrennungsverfahren untersucht.

- **Stoßfreie Explosionsverbrennung (*Shockless Explosion Combustion, SEC*):** Die sogenannte SEC ist ein zyklischer Prozess, der eine Verbrennung bei konstantem Volumen annähert. Die Theorie des Verfahrens wurde erstmals von Bobusch, Berndt, Paschereit und Klein, 2014 vorgestellt und von Reichel et al., 2016 experimentell untersucht. Seitdem wurden im Rahmen des Sonderforschungsbereichs 1029 zahlreiche theoretische und praktische Untersuchungen zu dem Konzept durchgeführt. Dabei ist das langfristige Ziel, die in Abb. 1.1 dargestellte Integration des Verbrennungsverfahrens in eine Gasturbine. Die stoßfreie Explosionsverbrennung (SEC) ist in vielerlei Hinsicht einem Pulsstrahltriebwerk ähnlich (Litke, Schauer, Paxson, Bradley & Hoke, 2005).

Der Ablauf eines SEC-Zyklus ist schematisch in Abb. 1.2 dargestellt. Gezeigt ist der

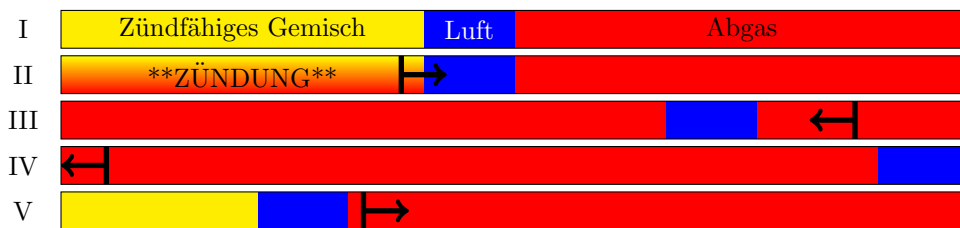


Abbildung 1.2: Schematische Darstellung eines SEC-Verbrennungsrohres über einen Zyklus

Querschnitt eines SEC-Verbrennungsrohres für verschiedene Phasen des Prozesses. Der Zyklus startet mit einer homogenen Selbstzündung des Brennstoff-Luft-Gemischs im vorderen Teil des Verbrennungsrohres (I). Durch die gleichmäßige Zündung im vorderen Teil des Rohres und dem damit verbundenen Druckanstieg bildet sich eine Druckfront aus, welche stromabwärts läuft (II). Sobald diese Druckfront das offene Ende des Rohres erreicht, wird sie reflektiert und läuft als Saugwelle zurück in Richtung des Rohreinlasses (III). Die Saugwelle lässt bei Erreichen des Rohreinlasses (IV) den Druck im Rohr unter den Versorgungsdruck vor dem Rohr fallen und ermöglicht so eine Wiederbefüllung des Rohres mit Frischluft. Während der Wiederbefüllung wird am Rohreinlass Kraftstoff eingespritzt (V). Um erneut eine homogene Selbstzündung im nächsten Zyklus zu realisieren, muss die Kraftstoffkonzentration über die Zeit der Einspritzung so variiert werden, dass am Ende der Wiederbefüllung das Kraftstoff-Luft-Gemisch im vorderen

Teil des Rohres passend geschichtet ist. Danach beginnt der Prozess erneut.

Der SEC-Zyklus wird durch die Anpassung von reaktions- und strömungsspezifischen Zeitkonstanten bestimmt. Unter der Annahme eines näherungsweise konstanten Versorgungsdrucks vor dem SEC-Rohr, ist die einzige Möglichkeit auf den Verbrennungsprozess Einfluss zu nehmen, die Einspritzung am Rohreinlass. Das durch die Kraftstoffschichtung definierte Zündverhalten bestimmt das Druckprofil im Rohr nach der unmittelbaren Verbrennung und die daraus resultierende Strömung im Rohr. Das Regelungsziel, welches durch die Einspritzung realisiert werden soll, bezieht sich also nicht nur auf die Verbrennung selbst, sondern auch auf das interagierende Strömungssystem.

- **Pulsierende Detonationsverbrennung (*Pulse Detonation Combustion*, PDC):**

Die Idee einer Detonationsverbrennung wurde ursprünglich in Zel'dovich, 1940 vorgestellt. Ausgehend davon wurde das Konzept in Nicholls, Wilkinson und Morisson, 1957 experimentell untersucht und danach vor allem als Antriebsmöglichkeit für Raketen weiterentwickelt (Endo et al., 2004; Kasahara, Hirano, Matsuo, Daimon & Endo, 2009). Zwar gibt es einige Untersuchungen zur Kombination mit einer Turbine (Caldwell, Glaser, Dimicco & Gutmark, 2005; Rouser, King, Schauer, Sondergaard & Hoke, 2010), allerdings konnte kein Aufbau über den Prototypenstatus hinaus betrieben werden.

In einer konventionellen Gasturbine breiten sich die Flammen im Unterschallbereich aus. Dieser Vorgang wird als Deflagration bezeichnet. Bei einer Detonation hingegen bildet sich eine Detonationsfront aus, welche sich mit Überschallgeschwindigkeit durch das Medium bewegt. In der pulsierenden Detonationsverbrennung wird in einem Rohr eine Detonationsfront erzeugt, welche sich so schnell ausbreitet, dass unverbranntes Gas vor der Detonationsfront kaum expandieren kann. Somit erfolgt der Energieumsatz in der Detonationsfront bei annähernd konstantem Volumen. Nach einem Verbrennungszyklus muss das Verbrennungsrohr gespült werden, bevor es erneut mit Luft und Kraftstoff beladen werden kann. Ein ausführlicher Überblick des PDC-Konzepts ist in Kailasanath, 2003 und Bussig und Pappas, 1996 zu finden.

In einer praktischen, technischen Umsetzung ist es unmöglich, einem detonationsfähigen Gemisch durch eine externe Zündquelle ausreichend Energie zuzuführen, um direkt eine Detonation zu initiieren. Daher entsteht in der Praxis eine Transitionsphase zwischen Deflagration und Detonation. Dieser Übergang kann durch bestimmte stoßfokussierende Geometrien im Verbrennungsrohr begünstigt werden. Details über die technische Umsetzung der PDC im Rahmen des Sonderforschungsbereichs 1029 sind in Gray et al., 2016, Haghdooost, Edgington-Mitchell, Paschereit und Oberleithner, 2019 und Habicht, Yücel, Hanraths, Djordjevic und Paschereit, 2021 zu finden.

Beide Verbrennungskonzepte führen sowohl am Einlass als auch am Auslass der Verbrennungsrohre zu Druckschwankungen. Um diese Druckschwankungen zu kompensieren, sind, wie in Abb. 1.1 gezeigt, Plena vor und hinter den Brennkammern notwendig. Damit soll verhindert werden, dass die mitunter starken Druckschwankungen den Betrieb der anliegenden Komponenten beeinträchtigen oder diese im schlimmsten Fall sogar beschädigen. Der druckerhöhende Charakter der Verbrennung sorgt dafür, dass zwischen den Plena vor und hinter der Brennkammer verschiedene Druckniveaus herrschen. Dieser Aufbau resultiert in diversen Herausforderungen, die zum Teil mit den Methoden der aktiven Strömungsbeeinflussung adressiert werden können. Die wichtigsten Aufgaben sind im Folgenden zusammen mit den grundlegenden, im Sonderforschungsbereich 1029 angestrebten Lösungsansätzen aufgelistet:

1 Einleitung

- Druckspitzen im Plenum hinter den Verbrennungsrohren können einen gleichmäßigen Betrieb der Turbine erschweren und bei zu großen Schwankungen diese sogar dauerhaft schädigen (Rouser et al., 2010). Zur Kompensation der Druckschwankungen, die aus den einzelnen Verbrennungsrohren in das Plenum eintreten, ist eine Synchronisation der Feuerung der einzelnen Rohre vorgesehen. So können Druckspitzen gegen Drucktäler ausgeglichen werden, um insgesamt möglichst gleichmäßige Bedingungen vor der Turbine sicherzustellen.
- Druckschwankungen im Plenum vor den Verbrennungsrohren können den Betrieb des Kompressors beeinflussen. Insbesondere besteht die Gefahr, den Kompressor in einen schädlichen Pumpbetrieb zu bringen. Um eine zuverlässige Kompression zu gewährleisten, sind in der letzten Statorstufe aktive Bauelemente vorgesehen, über die Luft in die Strömung eingeblasen werden kann. Damit können Störungen, die durch die zyklische Befüllung der Verbrennungsrohre entstehen, kompensiert und eine gleichbleibende Kompression garantiert werden. In der vorliegenden Arbeit wird dieser Aspekt nicht weiter betrachtet. Ansätze sind beispielsweise in Steinberg, Staats, Nitsche und King, 2017 und Fietzke, Mihalyovics, King und Peitsch, 2021 zu finden.
- Der zyklische Betrieb der SEC ist abhängig von einer präzisen Kraftstoffsichtung über die Länge der einzelnen Verbrennungsrohre. Dafür ist eine exakte Anpassung der eingespritzten Kraftstoffmenge am Rohreinlass notwendig. Dabei müssen bei der Befüllung des jeweiligen Rohres die aktuellen Strömungsverhältnisse sowohl im Rohr als auch davor berücksichtigt werden. Nur so kann der Kraftstoffstrom über die Dauer der Einspritzung optimal angepasst werden.
- Die Wiederbefüllung der Rohre ist aufgrund der unterschiedlichen Druckniveaus vor und hinter den Brennröhrn besonders schwierig. Für die Wiederbefüllung wird, wie im Konzept der SEC beschrieben, der Effekt der zurücklaufenden Saugwelle ausgenutzt. Um den Befüllungsprozess nicht zu behindern, muss sichergestellt sein, dass ein Druckabfall im Plenum nicht die Wirkung der Saugwelle im Verbrennungsrohr aufhebt. Dafür kann zum einen gezielt die Aktuation in der letzten Statorstufe genutzt werden. Zusätzlich ist es auch möglich, durch die Synchronisation der Feuerrohre und eine daraus resultierende Anpassung des Druckfeldes im Plenum sicherzustellen, dass die Befüllung zu einem passenden Zeitpunkt realisiert wird.

1.2 Motivation

Das Beispiel einer Gasturbine mit Gleichraumverbrennung verdeutlicht, wie vielfältig die Anforderungen der aktiven Strömungsbeeinflussung sein können. Im Allgemeinen lässt sich die Problematik des Aufbaus einer Regelung der genannten sowie allgemein komplexer Strömungsphänomene grob in zwei Bereiche unterteilen, welche die Schwerpunkte dieser Arbeit bilden:

1.2.1 Modellierung

Leistungsfähige Regelungsansätze, wie z.B. die modellprädiktive Regelung (MPC), aber auch Zustandsschätzverfahren als Grundlage für modellbasierte Regler oder zur Überwachung, erfordern eine möglichst akkurate mathematische Modellierung des zu regelnden Prozesses. Für

strömungsmechanische Systeme ist eine solche mathematische Modellierung durch die Navier-Stokes-Gleichungen bzw. für viele Prozesse in guter Näherung durch die Euler-Gleichungen möglich, gegebenenfalls ergänzt um Gleichungen, welche chemische Reaktionen berücksichtigen. Die Problematik dieser Modellierungen ist der mitunter hohe Rechenaufwand, der bei einer numerischen Lösung in Abhängigkeit der Diskretisierung von Ort und Zeit entsteht. Je nach Auflösung kann der Zeitaufwand für eine numerische Lösung selbst auf Hochleistungsrechnern um viele Größenordnungen über der realen Zeitskala des Prozesses liegen. Dies gilt insbesondere für zufriedenstellend genaue Lösungen von kompressiblen Strömungen mit Stoßphänomenen, wie sie im beschriebenen Konzept einer Gasturbine mit Gleichraumverbrennung zu erwarten sind. Dieser enorme Rechenaufwand numerischer Lösungen verhindert zumeist eine Verwendung der Modelle in optimierungsbasierten Regelungen wie der genannten MPC. Ansätze aus der Modellreduzierung bieten hier nur bedingt Abhilfe, da sie nur für eine oder eine Menge von bekannten Anfangs- und Randbedingungen in einem festen Raum-Zeit-Intervall gute Näherungen der echten Modelle garantieren.

Aufgrund dieser Schwierigkeiten in der physikalischen Modellierung wird oft auf verschiedene Methoden der Blackbox-Modellierung zurückgegriffen, um aus Messdaten ein praktikables Modell anzupassen. Diese Ansätze können mitunter gute Ergebnisse liefern, bergen jedoch das Risiko, in Situationen, die nicht in den Messdaten zur Anpassung repräsentiert sind, unvorhersehbares Verhalten zu zeigen. Zudem kann die fehlende physikalische Interpretierbarkeit der gewählten Zustände die Nutzbarkeit des Modells stark einschränken.

Der erste Teil dieser Arbeit beschäftigt sich daher mit einer neuen Methode zur Modellierung strömungsmechanischer Prozesse, welche speziell für die Verwendung in einer Regelung bzw. Zustandsschätzung entwickelt wurde. Dabei liegt der Fokus vor allem auf der Möglichkeit einer Auswertung in Echtzeit. Der vorgestellte Ansatz beruht dabei auf sogenannten *Physics-Informed Neural Networks* (PINN), die als Alternative oder Ergänzung zu numerischen Lösern entwickelt wurden. Ausgehend vom allgemeinen PINN-Konzept wird ein Zustandsraummodell eingeführt, welches sowohl für modellgestützte Messverfahren als auch in optimierungsbasierten Regelungsansätzen verwendet werden kann. Der Rechenaufwand in der Auswertung dieser Modelle liegt bei zufriedenstellender Genauigkeit Größenordnungen unter dem einer vollständigen, numerischen Lösung und ermöglicht damit bei passender Implementierung Regelungs- und Zustandsschätzanwendungen im Bereich der Echtzeitfähigkeit.

1.2.2 Regelung

Neben der Problematik, ein passendes und rechenzeiteffizientes Modell zu definieren, treten auch in der Reglersynthese für strömungsmechanische Systeme spezielle Herausforderungen auf. Die gravierendste ist in vielen Fällen eine mechanische Beschränkung der Aktuation. Die Dynamik der zu regelnden Prozesse erfordert eine hohe Aktuationsbandbreite, die von Bauteilen mit kontinuierlichem Charakter, wie z.B. konventionellen Proportionalventilen, nicht geliefert werden kann. Dies gilt insbesondere, wenn für die Aktuation relativ große Massenströme benötigt werden. Höhere Bandbreiten in der Aktuation können beispielsweise durch schaltbare Elemente realisiert werden. Allerdings ist auch deren Bandbreite oftmals nicht ausreichend, um mit einem hochfrequenten, pulsweitenmodulierten Signal angesteuert zu werden und damit eine Reglersynthese in einem kontinuierlichen Bereich zu ermöglichen. Folglich muss die Reglersynthese direkt für einen diskreten Wertebereich erfolgen, welcher die tatsächlichen Aktuationsmöglichkeiten repräsentiert.

Da bisher nur wenige Ansätze für die Reglersynthese in einem binären oder ganzzahligen

Wertebereich existieren, werden im zweiten Teil dieser Arbeit verschiedene Ansätze, welche derart beschränkte Aktuation ermöglichen, vorgestellt und diskutiert. Im Hinblick auf den zyklischen Charakter vieler strömungsmechanischer Systeme, insbesondere im Falle des beschriebenen Gasturbinenkonzepts, ist dieser Teil der Arbeit vor allem auf die Reglerynthese für eben diese zyklischen Systeme fokussiert. Insbesondere liegt der Schwerpunkt auf einem speziellen Syntheseverfahren für eine ganzzahlige, iterativ lernende Regelung (ILR), sowie einem modellfreien Regelungsansatz, der sich an der Idee der Extremwertregelung (EWR) orientiert.

1.3 Aufbau der Arbeit

Die vorliegende Arbeit unterteilt sich in die zwei oben vorgestellten Themenkomplexe. Im ersten Teil der Arbeit wird auf die Modellierung von Strömungssystemen eingegangen und eine Methode der Modellapproximation vorgestellt, die speziell für die Verwendung in Algorithmen der Zustandsschätzung bzw. der modellbasierten Regelung angepasst ist. Im zweiten Teil wird auf die Regelung von zyklischen Prozessen eingegangen, deren Aktuation auf einen ganzzahligen Wertebereich beschränkt ist. Thematisch sind die Kapitel wie folgt unterteilt:

- **Kapitel 2** gibt einen Überblick über den Stand der Forschung der verschiedenen Forschungsgebiete, aus denen sich diese Arbeit zusammensetzt. Es wird kurz auf die Modellierung von Strömungsphänomenen und das Feld der beschränkten, optimierungsbasierten Regelung eingegangen. Zusätzlich werden die gängigen Methoden und Anwendungen der aktiven Strömungsbeeinflussung zusammengefasst.
- **Kapitel 3** führt zunächst die gängigen mathematischen Modelle zur Beschreibung von Strömungssystemen ein. Ausgehend von den allgemeinen Navier-Stokes-Gleichungen werden die Euler-Gleichungen bzw. die Burgers-Gleichung eingeführt. Zudem werden gängige Methoden zur Modellreduktion vorgestellt. Im Anschluss wird kurz auf die Theorie neuronaler Netze eingegangen, bevor das spezielle Konzept der erwähnten PINNs vorgestellt wird. Davon ausgehend wird eine Methode der Zustandsraummodellierung auf Basis neuronaler Netze präsentiert. Der Ansatz wird dann beispielhaft auf die Burgers-Gleichung und die Euler-Gleichungen angewendet. Eine Applikation an realen Komponenten einer Gleichraumgasturbine kann nicht erfolgen, da die hierzu notwendigen, experimentellen Einrichtungen im Sonderforschungsbereich 1029 fehlen. Gleichwohl können die hier entwickelten Ansätze für zukünftige Arbeiten zum Einsatz kommen.
- **Kapitel 4** zeigt die beispielhafte Verwendung der Modelle aus Kapitel 3 in einem Zustandsschätz-Szenario. Dabei wird ein erweitertes Kalman-Filter verwendet und auf den Einsatz für Systeme auf Basis der Burgers-Gleichung sowie der Euler-Gleichungen eingegangen.
- **Kapitel 5** zeigt analog zum vorangegangenen Kapitel die Verwendung der Zustandsraummodelle für Burgers-Gleichung und Euler-Gleichungen in einem Regelungskontext. Dabei werden die Modelle in eine MPC integriert.
- **Kapitel 6** geht detailliert auf die Theorie und Anwendung der ganzzahlig, beschränkten ILR ein. Im ersten Schritt werden ausgehend von der reellwertigen, unbeschränkten ILR die Konvergenzeigenschaften der ganzzahligen ILR diskutiert und Anforderungen an eine ganzzahlige ILR formuliert. Im Anschluss wird ein besonders zielführendes und

recheneffizientes Syntheseverfahren für eine ganzzahlige ILR vorgestellt und mit vorhandenen Methoden verglichen. Das eingeführte Syntheseverfahren wird anschließend genutzt, um einen Regler für eine einfache Simulationsstudie aufzubauen, in welcher die verschiedenen Ansätze verglichen werden. Im zweiten Teil des Kapitels wird genauer auf den Sonderfall einer binären ILR eingegangen. Es wird der Vergleich zwischen einer optimierungsbasierten und einer rundungsbasierten Methode gezogen. Darauf folgend wird ein spezielles Verfahren für eine binäre ILR vorgestellt, welches besonders recheneffizient für Anwendungen mit hochdimensionalen Stellfolgen genutzt werden kann. Die Methode wird abschließend an einem akustischen Ersatzversuchsaufbau getestet, der das Plenum einer Gasturbine nach der Gleichraumverbrennung imitiert, siehe Abb. 1.1.

- **Kapitel 7** präsentiert einen alternativen Ansatz für die Regelung von zyklischen Prozessen, in denen die Aktuation auf einen ganzzahligen Wertebereich beschränkt ist. Der Ansatz basiert dabei auf dem Konzept der EWR und wird im Anschluss für die Auslegung einer Regelung in einer SEC-Simulationsstudie genutzt. Zusätzlich wird an dieser Stelle der Vergleich zu einer ganzzahligen ILR gezogen. Im Anschluss wird die zyklische EWR zur Anpassung der Kraftstoffeinspritzung an einem SEC-Experiment genutzt, um den aus der Verbrennung resultierenden Druckanstieg zu optimieren.
- **Kapitel 8** fasst die Ergebnisse dieser Arbeit zusammen und gibt einen kurzen Ausblick zu den in dieser Arbeit vorgestellten Methoden.

2 Stand der Forschung

In diesem Abschnitt soll ein Überblick über relevante Arbeiten gegeben werden, die sich mit einem oder mehreren Teilaspekten dieser Dissertation befassen. Im ersten Abschnitt wird ein Überblick über Modellierungsmethoden strömungsmechanischer Systeme gegeben. Dabei liegt der Fokus vor allem auf dem Rechenaufwand und der Verwendung der einzelnen Methoden in einem Regelungskontext. Der zweite Abschnitt wird auf Arbeiten aus dem Bereich der beschränkten Regelung eingehen, bevor der letzte Abschnitt einen kurzen Überblick über die aktuelle Entwicklungen auf dem Feld der aktiven Strömungsbeeinflussung gibt.

2.1 Strömungsmodellierung

Für die mathematische Modellierung von Strömungssystemen haben sich je nach Komplexität des Prozesses verschiedene Konzepte etabliert. Der Zusammenhang zwischen den im Folgenden aufgeführten Modellen zur Beschreibung eines Strömungsfeldes ist ohne Anspruch auf Vollständigkeit in Abb. 2.1 dargestellt. Grundsätzlich können die Modelle in die Gruppen

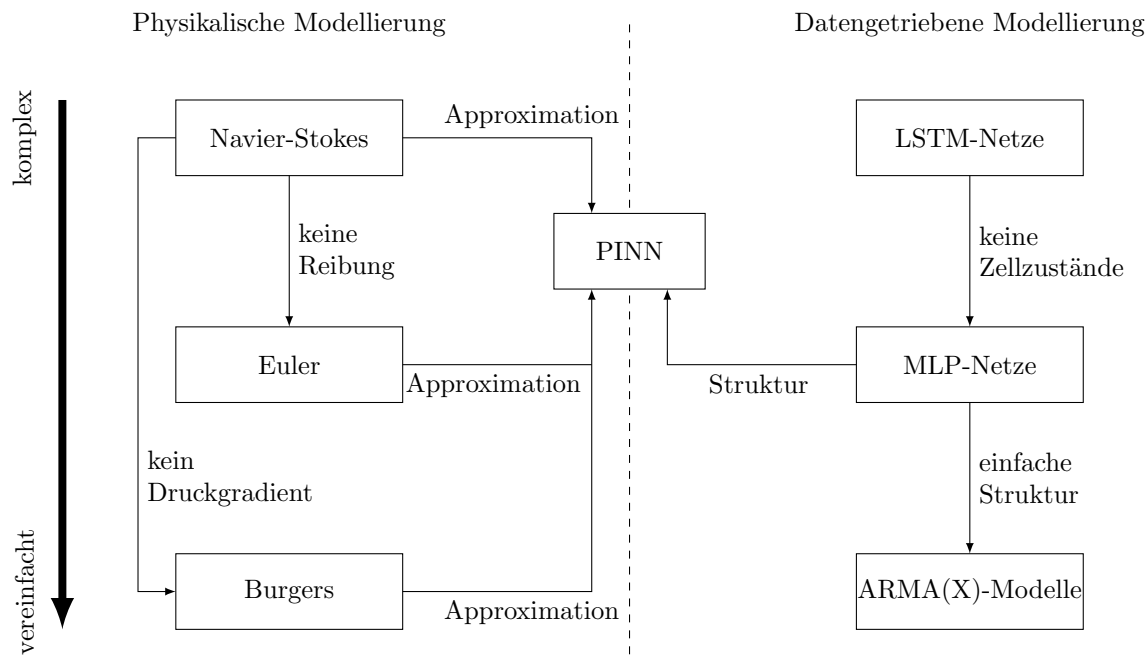


Abbildung 2.1: Zusammenhang zwischen verschiedenen Modellen zur Beschreibung von Strömungssystemen.

physikalische Modelle und datengetriebene Modelle unterteilt werden.

Die allgemeinste Beschreibungsform eines Strömungsfeldes, die auf physikalischen Grundannahmen aufbaut, sind die Navier-Stokes-Gleichungen (Lemarie Rieusset, 2016; Robinson, Rodrigo & Sadowski, 2016). Die Navier-Stokes-Gleichungen beschreiben, ausgehend von der

2 Stand der Forschung

Impulserhaltung für ein Kontinuum, Strömungen in viskosen, kompressiblen oder inkompressiblen newtonschen Fluiden, gegebenenfalls erweitert um Gleichungen zur Beschreibung von Reaktionen. Damit lässt sich ein breites Spektrum an technisch relevanten Systemen abbilden, wie z.B. die Umströmung von starren Körpern oder Wirbelphänomene. Aus diesem Grund werden die Navier-Stokes-Gleichungen heute in vielen verschiedenen Bereichen wie dem aerodynamischen Design von Bauteilen (Dhert, Ashuri & Martins, 2017) oder der Wetterforschung (Kimura, 2002) verwendet. Auf die genaue mathematische Beschreibung wird im folgenden Kapitel näher eingegangen. Da eine analytische Lösung der Navier-Stokes-Gleichungen bisher nur für einige Sonderfälle angegeben werden kann (Ramm, 2017), bedarf es numerischer Methoden, um eine Anwendung in der Praxis zu ermöglichen. Für diese Art von partiellen Differentialgleichungen hat sich vor allem die Finite-Volumen-Methode (FVM) etabliert. Ein allgemeiner Überblick über die FVM kann z.B. in Eymard, Gallouet und Herbin, 2000 gefunden werden. Die Grundlagen des Ansatzes sind zudem in Anhang A erklärt. Eine ausführliche Erklärung der FVM mit dem Fokus auf die Verwendung in der Fluid-dynamik ist in Moukalled, Mangani und Darwish, 2015 zu finden. Obwohl die Lösung der Navier-Stokes-Gleichungen mit geeigneten numerischen Methoden gute Übereinstimmung mit Experimenten zeigt, ist in der Anwendung aufgrund des hohen Rechenaufwands oftmals eine Abwägung zwischen gewünschter Genauigkeit und benötigter Rechenzeit notwendig. Durch die große Rechenzeit, die für die numerische Lösung benötigt wird, ist eine Verwendung für Echtzeitanwendungen wie Regelung oder Zustandsschätzung zumeist nicht möglich.

Da sich für viele technische Prozesse vereinfachende Annahmen treffen lassen, haben sich neben den Navier-Stokes-Gleichungen auch reduzierte mathematische Beschreibungen für Strömungssysteme etabliert. Werden z.B. Reibungseinflüsse vernachlässigt, ergeben sich die Euler-Gleichungen, die in dieser Arbeit schwerpunktmäßig verwendet werden. Zwar existieren für die Euler-Gleichungen unter bestimmten Bedingungen Lösungsverfahren, z.B. über die Methode der Charakteristiken (Roe, 1986), in den meisten Fällen muss jedoch auch hier die Lösung über die FVM numerisch bestimmt werden. Zudem existieren diverse Erweiterungen z.B. mit künstlichen Verlusten (Pulliam, 1986), welche die Euler-Gleichungen um bestimmte Eigenschaften ergänzen, ohne auf die vollständigen Navier-Stokes-Gleichungen zurück zu kommen. Die Euler-Gleichungen stellen zudem die Grundlage dar, auf denen im Sonderforschungsbereich 1029 zahlreiche numerische Untersuchungen der SEC durchgeführt wurden (Berndt, 2014; Berndt & Klein, 2016; Schäpel et al., 2017).

Eine deutlich vereinfachte Beschreibung von Strömungen bietet die Burgers-Gleichung. Sie bildet einfache inkompressible Strömungen ab. So können Systeme mit und ohne Viskosität betrachtet werden. Im Gegensatz zu den Navier-Stokes-Gleichungen und den Euler-Gleichungen ist die Burgers-Gleichung über die Methode der Charakteristiken (Shi, Zhou & Shi, 2013) oder mit der sogenannten Hopf-Colman-Transformation (Gorguis, 2006) analytisch lösbar. Die Burgers-Gleichung wird daher oft als Referenzbeispiel für numerische Lösungsverfahren herangezogen, da ein direkter Vergleich von numerischer und analytischer Lösung möglich ist. Auch in dieser Arbeit wird die vorgestellte Modellierungsmethode, bevor sie auf die Euler-Gleichungen angewendet wird, für die Burgers-Gleichung getestet.

Der hohe Rechenaufwand der numerischen Lösung der Navier-Stokes-Gleichungen und der Euler-Gleichungen erschwert die Nutzung in zeitkritischen Anwendungen. Um den Rechenaufwand zu reduzieren, bieten sich Verfahren der Modellreduktion an. Eine der verbreitetsten Methoden ist die sogenannte *Proper Orthogonal Decomposition* (POD). Dabei werden mit dem vollständigen Modell eine bestimmte Menge an Schnappschüssen des Raum-Zeit-Bereichs erstellt und für diese Menge die Hauptkomponenten bestimmt. Aus den Hauptkomponenten lässt sich anschließend ein Modell aufbauen, welches entsprechend der Anzahl der

berücksichtigten Hauptkomponenten mehr oder weniger stark reduziert ist. Sowohl für die Euler-Gleichungen (Willcox & Peraire, 2002; Serre, Lafon, Gloerfelt & Bailly, 2012) als auch für die Navier-Stokes-Gleichungen (Rowley, Colonius & Murray, 2004; Shinde, Longatte, Baj, Hoarau & Braza, 2016) lassen sich damit gute Ergebnisse erzielen. Zudem gibt es diverse Erweiterungen des POD-Ansatzes wie z.B. die shifted POD (Reiss, Schulze, Sesterhenn & Mehrmann, 2018), die für bestimmte Systeme die Ergebnisse verbessern können. Ein kurzer Überblick über die gängigsten Methoden der Modellreduktion ist in Abschnitt 3.1.4 zu finden. Eine Schwierigkeit bei der Modellreduktion ist der Erhalt bestimmter Systemeigenschaften wie z.B. der Stabilität (Iollo, Lanteri & Désidéri, 2000). Zudem ist eine direkte Übertragung, beispielsweise in ein Zustandsraummodell nicht unmittelbar möglich, da das reduzierte Modell nur in einem festen Zeitbereich gültig ist. Alternativen zu klassischen Verfahren der Modellreduktion bieten Methoden des maschinellen Lernens, mit dessen Hilfe z.B. die Lösung der partiellen Differentialgleichungen approximiert wird. Nach ersten Ergebnissen für die Navier-Stokes-Gleichungen in Lomakin und Korsunov, 2019, hat sich der Ansatz der sogenannten PINN (Raissi, 2018; Raissi, Perdikaris & Karniadakis, 2019) in den letzten Jahren als besonders praktikabel erwiesen. Dabei wird ein einfaches *Multilayer Perceptron* (MLP) verwendet, um die Lösung der partiellen Differentialgleichung zu approximieren. Im Trainingsprozess des MLPs wird neben bekannten Lösungen der partiellen Differentialgleichung, wie Anfangs- und Randbedingungen, auch der Zusammenhang zwischen den Ableitungen gemäß der partiellen Differentialgleichung berücksichtigt. Das Konzept der PINNs wird in Abschnitt 3.3 ausführlich erläutert. Auch wenn dieser Ansatz gute Ergebnisse bei vergleichsweise moderatem Rechenaufwand zeigt, werden in den genannten Veröffentlichungen ausschließlich Lösungen für autonome Systeme auf festen Raum-Zeit-Intervallen betrachtet. Damit ist die direkte Verwendung in einem für die Regelungstechnik geeigneten Zustandsraummodell nicht möglich.

Wie in Abb. 2.1 angedeutet, existieren neben den Strömungsmodellen, die auf der physikalischen Beschreibung eines Kontinuums aufbauen, verschiedenste Verfahren der datengetriebenen Modellbildung. Voraussetzung für eine datengetriebene Modellierung ist eine ausreichende Menge an zeitlich aufgelösten Eingangs- und Ausgangsdaten des zu beschreibenden Systems. Ein einfacher und immer noch häufig verwendeter Ansatz für die Zeitreihenbeschreibung ist die Berechnung des Ausgangssignals des folgenden Zeitschritts als eine Linearkombination vergangener Eingangs- und Ausgangsdaten (Choi, 2012). Diese Modelle werden meistens unter dem Begriff *Autoregressive Moving Average* (ARMA(X))-Modelle zusammengefasst. Das (X) steht hier für einen möglichen exogenen Eingang. Dabei haben sich diverse Kombinationen als besonders praktikabel erwiesen. Ein Überblick ist in (T. W. Liao, 2005) zu finden. Neben der Linearkombination gibt es auch Ansätze für eine nichtlineare Abbildung von zurückliegenden Daten auf den Ausgang im folgenden Zeitschritt (Parlitz, 1998). Dazu zählen auch einfach strukturierte neuronale Netzwerke wie MLPs (Ahmed, Atiya, El Gayar & El-Shishiny, 2010). Auch wenn sich bereits mit einfacheren Netzwerkstrukturen mitunter Modelle mit guter Approximationsgenauigkeit ergeben, haben sich in den letzten Jahren vor allem sogenannte *Long Short-Term Memory* (LSTM)-Netze für die Zeitreihenvorhersage etabliert (Lindemann, Müller, Vietz, Jazdi & Weyrich, 2021). LSTM-Netze werden mittlerweile neben der Strömungsmodellierung (Hasegawa, Fukami, Murata & Fukagata, 2020) auch für die Modellierung des Wetters (Karevan & Suykens, 2020) oder zur Vorhersage von Finanzdaten (Kim & Won, 2018) verwendet.

Neben den genannten Modellierungsverfahren gibt es unzählige Ansätze, welche speziell für einzelne technische Systeme entwickelt wurden. Dabei handelt es sich sowohl um physikalisch und empirisch motivierte Modelle als auch um sogenannte hybride Modelle, die beide Ansätze

ze vereinen. Ein umfangreicher Überblick zu den Modellierungsansätzen, die beispielsweise speziell für Turbomaschinen verwendet werden, ist in Horlock und Marsh, 1971 zu finden.

2.2 Regelung mit beschränkter Stellgröße

Beschränkungen der Stellgröße treten für viele Regelungsaufgaben auf. Sie resultieren zu meist aus einer technischen Limitierung können aber auch zur allgemeinen Reduzierung der Stellenergie verwendet werden. Der Umgang mit solchen Beschränkungen ist je nach verwendetem Verfahren für die Reglersynthese unterschiedlich. Daher wird an dieser Stelle vor allem auf Beschränkungen in modellgestützten, optimierungsbasierten Ansätzen sowie auf den speziellen Fall der ganzzahligen Beschränkung der Stellgröße eingegangen.

Regelungsansätze wie die MPC (Morari, Garcia & Prett, 1988; Qin & Badgwell, 2003) oder die norm-optimale ILR (Amann, Owens & Rogers, 1998) führen in der Reglersynthese für lineare bzw. linearisierte System häufig zu einem quadratischen Optimierungsproblem, ein sogenanntes quadratisches Programm (QP). In einem QP können lineare Gleichungs- und Ungleichungsbeschränkungen berücksichtigt werden. Damit lassen sich viele technisch relevante Begrenzungen, wie z.B. obere und untere Schranken für die Stellgröße, in der Regelung berücksichtigen. Zur Lösung von QPs haben sich vor allem die Innere-Punkt-Methode (Pólik & Terlaky, 2010) und die Methode der aktiven Menge (Forsgren, Gill & Wong, 2016) etabliert. Eine theoretische Betrachtung der beiden Methoden im Hinblick auf eine beschränkte ILR ist in Mishra, Topcu und Tomizuka, 2011 zu finden. Peng, Sun, Zhang und Tomizuka, 2016 zeigt ein anschauliches Beispiel einer Temperaturregelung auf Basis einer beschränkten ILR.

Auch wenn sich Beschränkungen einfach in einem QP berücksichtigen lassen, können sie mitunter die Stabilität und das Konvergenzverhalten der Regelung maßgeblich stören. Besonders im Bereich der MPC gibt es umfangreiche Untersuchungen zu den Auswirkungen von Beschränkungen auf Stabilität und Konvergenz (Mayne, Rawlings, Rao & Sckaert, 2000).

Alternativ zu den Ansätzen über QP-Lösungsverfahren existieren für die ILR auch Ansätze, um Beschränkungen im System über Projektionsmethoden zu berücksichtigen. Dabei können z.B. Energiebeschränkungen oder obere und untere Schranken für die Stellgröße berücksichtigt werden (Chu & Owens, 2010). Konvergenzanalysen für entsprechende Regelungen sind jedoch zumeist nur problemspezifisch, wie z.B. in Gluck, Blank, Buchl und Kugi, 2018 möglich. Auch für den modellfreien Ansatz der EWR existieren Syntheseverfahren auf Basis einer Projektion (Mills & Krstic, 2014), in denen Beschränkungen berücksichtigt werden können. Alternativ existieren für die beschränkte EWR auch Verfahren, denen ein nichtlinearer Optimierungsansatz zu Grunde liegt (C.-K. Liao, Manzie, Chapman & Alpcan, 2019).

Eine Beschränkung der Aktuation auf einen ganzzahligen Bereich ist, im Vergleich zu linearen Beschränkungen, ein deutlich größeres Problem für die Berücksichtigung in einer Regelung. Zwar existieren Lösungsverfahren für ganzzahlige QPs, allerdings hat diese Art der Beschränkung einen deutlich größeren Einfluss auf Konvergenz und Stabilität. Hinzu kommt, dass der Rechenaufwand für ganzzahlige QPs deutlich größer ist als für eine reellwertige Lösung. Trotz dieser Umstände gibt es einige wenige Arbeiten zu ganzzahligen Regelungsansätzen. In Bemporad und Morari, 1999 und Bürger, Zeile, Altmann-Dieses, Sager und Diehl, 2019 werden MPC-Ansätze für logische bzw. schaltende Systeme vorgestellt und auf Basis einer ganzzahligen Optimierung gelöst. Axehill, Vandenberghe und Hansson, 2010 betrachten eine MPC

mit einem allgemeinen, ganzzahligen Eingang. Zudem wurde im Rahmen des SFB 1029 eine erste Arbeit zu einer ganzzahligen ILR vorgestellt (Schäpel et al., 2018).

2.2.1 Ganzzahlige quadratische Optimierung

Da in dieser Arbeit an vielen Stellen auf die Lösung ganzzahliger QPs verwiesen wird, soll an dieser Stelle ein kurzer Überblick über den Stand der Technik zur Lösung dieser Optimierungsprobleme gegeben werden.

Der Hauptunterschied eines ganzzahligen QP gegenüber dem reellwertigen Fall liegt in der Unterteilung des Lösungsraums in diskrete Bereiche. Die endliche Anzahl an zulässigen Lösungen muss folglich mit einem möglichst effizienten Algorithmus durchsucht werden, um die optimale Lösung zu finden. Ein erster Ansatz über einen Suchbaum ist in Dakin, 1965 zu finden. Über die Jahre wurde der Suchbaum-Algorithmus zwar immer wieder in Details angepasst, der zugrundeliegende Ansatz wird aber bis heute angewendet. Aktuelle Lösungsverfahren für ganzzahlige QPs basieren auf dem sogenannten „*Branch-and-Bound*“-Algorithmus. Ein Überblick über diese Art der Baumsuche ist in Morrison, Jacobson, Sauppe und Sewell, 2016 zu finden. In vielen Fällen wird der Algorithmus durch ein Schnittebenenverfahren ergänzt, um die zulässigen Lösungen schneller einzuschränken. Informationen zum Schnittebenenverfahren für ganzzahlige Optimierung sind in Marchand, Martin, Weismantel und Wolsey, 2002 zu finden. Mit dieser Erweiterung wird zumeist vom „*Branch-and-Cut*“-Verfahren gesprochen.

Der Rechenaufwand des eigentlichen Suchverfahrens wird maßgeblich durch zwei Faktoren beeinflusst:

- Da in jedem Explorationsschritt des Suchbaums ein reellwertiges, beschränktes QP gelöst werden muss, hat der dafür verwendete QP-Löser einen großen Anteil an der effizienten Umsetzung des Suchverfahrens. Für die Verwendung in ganzzahligen QP-Lösern hat sich vor allem die Methode der aktiven Menge etabliert (Ferreau, Kirches, Potschka, Bock & Diehl, 2014). Speziell die sogenannte *Primal*- bzw. „*Dual Active Set*“-Methode (Forsgren et al., 2016) ist weit verbreitet. Der Vorteil der Methode der aktiven Menge liegt in der Möglichkeit, bei der Lösung ähnlicher QPs Informationen einer bereits erhaltenen Lösung wieder zu verwenden. Daher existieren Ansätze, die speziell auf die Anwendung in ganzzahligen Optimierungen angepasst sind (Buchheim, Santis, Lucidi, Rinaldi & Trieu, 2016). Ein Überblick über den Rechenaufwand der Methode ist in Cimini und Bemporad, 2017 zu finden.
- Mit einer entsprechenden Heuristik wird bestimmt, nach welcher Strategie der Suchbaum exploriert wird bzw. an welchen Stellen das Schnittebenenverfahren benutzt wird. Die Möglichkeiten durch Heuristiken den Ablauf der Suche zu beeinflussen, reichen von der Entscheidung zwischen Breiten- oder Tiefensuche über die Wahl der initialen Lösung bis hin zur Entscheidung, für welche Variable zusätzliche Beschränkungen eingeführt werden sollen. Die Wahl einer sinnvollen Heuristik kann dabei stark von der späteren Verwendung des Löser abhängen. So existieren z.B. Heuristiken speziell für die schwierige Integration in eingebetteten Systemen (Takapoui, Moehle, Boyd & Bemporad, 2020).

In der breiten Anwendung haben sich in den letzten Jahren vor allem die Software-Pakete CPLEX (Blik, Bonami & Lodi, 2014) und Gurobi (Gurobi Optimization, 2020) etabliert. Da die Effizienz der Löser sehr stark von der geschickten Anpassung von Heuristiken abhängt, sind über die genaue Umsetzung der Löser in den Software-Paketen keine Informationen

verfügbar. Eine Untersuchung des Rechenaufwands von Lösungsverfahren für ganzzahlige QPs ist in Martin, 2001 zu finden. Abgesehen von der direkten Verwendung zur Bestimmung von QP-Lösungen, existieren auch optimierte Verfahren für die Verwendung in allgemeinen nichtlinearen Problemstellungen (Vielma, Dunning, Huchette & Lubin, 2017).

2.3 Aktive Strömungsbeeinflussung

An dieser Stelle soll ein kurzer Überblick über ausgewählte, aktuelle Arbeiten im Bereich der aktiven Strömungsbeeinflussung gegeben werden. Auf Details der einzelnen Umsetzungen wird an dieser Stelle nicht eingegangen. Ein ganz allgemeiner Überblick über die verschiedenen Aspekte der aktiven Strömungsbeeinflussung ist in Collis, Joslin, Seifert und Theofilis, 2004 zu finden.

Wie oben bereits erwähnt, ist in vielen Fällen die technische Realisierung der Aktuation in der Strömungsbeeinflussung äußerst komplex. Daher ist die Umsetzung der Aktuation ein wichtiger Aspekt in der Forschung. Ein Überblick über gängige Aktuationsmethoden ist in Cattafesta und Sheplak, 2011 und Liu, Wang, Luo, Xia und Deng, 2012 zu finden.

Aktive Strömungsbeeinflussung wird in vielen Bereichen, vor allem aber zur Optimierung aerodynamischer Eigenschaften, erforscht. So existieren diverse Arbeiten zur Aktuation des Seitenruders an Flugzeugen (Whalen et al., 2018; Andino et al., 2019) durch eine Verminderung der Strömungsseparation. Aber auch Ansätze mit statischen und oszillierenden Jets wurden beispielsweise für Flaps an Flugzeugtragflügeln (Melton, Koklu, Andino & Lin, 2018) oder an Windrädern (Aubrun, Leroy & Devinant, 2017) untersucht. Ebenso gibt es Arbeiten mit dem Fokus auf Fahrzeugen. Dabei ist das Ziel, beispielsweise den Abtrieb während der Fahrt zu verringern (McNally et al., 2015).

Ein Großteil der Untersuchungen im Bereich aktive Strömungsbeeinflussung beschränkt sich auf die Verwendung eines offenen Regelkreises. Allerdings gibt es auch einzelne Arbeiten, die sich auf einen geschlossenen Regelkreis beziehen. Die Regelungsansätze reichen von klassischen, robusten Reglerauslegungen im Frequenzbereich über \mathcal{H}_∞ -Verfahren (Kiesner & King, 2015, 2017) bis zu modellgestützten Syntheseverfahren, wie der MPC (Krentel, Muminovic, Brunn, Nitsche & King, 2010) bzw. der repetitiven MPC (Steinberg et al., 2017) oder der ILR (Steinberg, King, Staats & Nitsche, 2015). An dieser Stelle muss allerdings daraufhin hingewiesen werden, dass in keinem der genannten Ansätze ein vollständiges, physikalisches Modell auf Basis der Euler- oder Navier-Stokes-Gleichungen verwendet wurde. Allerdings existieren einige Arbeiten, in denen ein reduziertes Modell verwendet wird (Aleksić et al., 2008) oder ausschließlich einzelne Aspekte, wie beispielsweise die Wirbel, in der Modellierung berücksichtigt werden (Pastoor, Henning, Noack, King & Tadmor, 2008). Neben diesen konventionellen Regelungsansätzen gibt es auch ein größer werdendes Bestreben darin, Methoden aus dem Bereich des maschinellen Lernens zur Strömungsbeeinflussung einzusetzen. Dabei ist das Ziel, das Stellgesetz selbst mithilfe entsprechender Algorithmen zu ersetzen und im besten Fall gänzlich auf ein Modell des Systems zu verzichten. Ein Überblick über diese Ansätze ist in Duriez, Brunton und Noack, 2017 oder Brunton, Noack und Koumoutsakos, 2020 zu finden.

Neben den genannten modellbasierten Methoden mit einfachen, empirischen Modellen, existieren auch einige Arbeiten, in denen eine volle physikalische Beschreibung in der Steuerungs- bzw. Reglerauslegung verwendet wurde. Vor allem Optimalsteuerungsansätze für Systeme, die durch die Navier-Stokes-Gleichungen beschrieben werden, wurden untersucht (Desai & Ito,

1994; Gunzburger & Manservigi, 2000). Für entsprechende Systeme wurde in R. Vazquez, Schuster und Krstic, 2007 eine mögliche Methode zur Stabilisierung vorgestellt. Bei den genannten Arbeiten auf Basis der physikalischen Modelle muss jedoch festgehalten werden, dass es sich dabei um rein theoretische Untersuchungen handelt, die nicht unmittelbar auf ein Experiment übertragen werden können.

3 Modellierung

Die Qualität einer modellbasierten Regelung sowie die Möglichkeit einer echtzeitfähigen Umsetzung sind abhängig von der Art der Modellierung bzw. der Genauigkeit des Modells. In diesem Abschnitt werden zunächst die im vorangegangenen Kapitel aufgeführten physikalischen Modellierungsmethoden für Strömungssysteme mittels partiellen Differentialgleichungen genauer erläutert und deren Verwendung in optimierungsbasierten Regelungsansätzen diskutiert. Danach wird auf eine datengetriebene Methode zur Approximation von Differentialgleichungslösungen mittels neuronaler Netze eingegangen. Auf Basis dieses Konzeptes wird im Anschluss ein Zustandsraummodell formuliert, welches für modellgestützte Mess- und Regelungsverfahren vorteilhaft verwendet werden kann. Im Gegensatz zur konventionellen Lösung partieller Differentialgleichungen zeichnet sich die vorgestellte Methode durch einen erheblich reduzierten Rechenaufwand aus. Am Ende des Kapitels wird die Methode genutzt, um exemplarisch ein Zustandsraummodell für ein System auf Basis der Burgers-Gleichung bzw. der Euler-Gleichungen zu erstellen.

Es soll an dieser Stelle betont werden, dass der vorgestellte Ansatz nicht das Ziel hat, vorhandene numerische Löser zu ersetzen oder eine Alternative für sie darzustellen. Vielmehr soll eine Möglichkeit aufgezeigt werden, wie ausgehend von einem komplexen und detaillierten physikalischen Modell eine Systembeschreibung bestimmt werden kann, die in einer Echtzeitanwendung genutzt werden kann.

3.1 Konventionelle Modellierungsansätze für Strömungen

Konventionelle Methoden der physikalischen Beschreibung von Strömungssystemen werden üblicherweise aus Kontinuumsannahmen hergeleitet. Auf die detaillierten Herleitungen der einzelnen Gleichungen wird an dieser Stelle verzichtet und auf die Ausführungen von Spurk, Schobeiri und Marschall, 2012 oder Oertel, Böhle und Reviol, 2015 verwiesen. Hier werden mit den Navier-Stokes-Gleichungen, den Euler-Gleichungen und der Burgers-Gleichung kurz die drei gängigsten partiellen Differentialgleichungen bzw. Systeme aus partiellen Differentialgleichungen aufgeführt. Abhängig von bestimmten Annahmen und der Vernachlässigung einzelner Effekte, lässt sich mit diesen drei Gleichungen die Dynamik einer Vielzahl von Strömungsphänomenen beschreiben.

3.1.1 Navier-Stokes-Gleichungen

Die umfassendste Beschreibungsmöglichkeit laminarer und turbulenter Strömungsfelder bieten die Navier-Stokes-Gleichungen. Das Gleichungssystem leitet sich aus der Annahme der Impulserhaltung für ein Kontinuum ab. Wie in Oertel et al., 2015 beschrieben, gilt für die Änderung des Impulses eines Kontinuums

$$\dot{I}_K = \sum \dot{I}_{\text{in}} - \sum \dot{I}_{\text{out}} + \sum F_S + \sum F_V \quad (3.1)$$

3 Modellierung

mit

- $\dot{I}_{\text{in}} \rightarrow$ Eintretende Impulsströme
- $\dot{I}_{\text{out}} \rightarrow$ Austretende Impulsströme
- $F_S \rightarrow$ Normal- bzw. Scherspannung
- $F_V \rightarrow$ Volumenkräfte .

In kartesischen Koordinaten $\mathbf{x} = [x, y, z]^T$ mit den entsprechenden, richtungsabhängigen Geschwindigkeiten $\mathbf{v} = [v_x, v_y, v_z]^T$ sowie der dynamische Viskosität κ und der Dichte ρ ergibt sich aus der Impulserhaltung im kompressiblen Fall

$$\rho \left(\frac{\partial \mathbf{v}}{\partial t} + (\mathbf{v} \cdot \nabla) \mathbf{v} \right) = \mathbf{f} - \nabla p + \kappa \Delta \mathbf{v} + \frac{1}{3} \kappa \nabla (\nabla \cdot \mathbf{v}) \quad . \quad (3.2)$$

Dabei gilt

$$\nabla p = \left[\frac{\partial p}{\partial x}, \frac{\partial p}{\partial y}, \frac{\partial p}{\partial z} \right]^T \quad (3.3)$$

$$\mathbf{v} \cdot \nabla = v_x \frac{\partial}{\partial x} + v_y \frac{\partial}{\partial y} + v_z \frac{\partial}{\partial z} \quad (3.4)$$

$$\Delta \mathbf{v} = \frac{\partial^2 \mathbf{v}}{\partial x^2} + \frac{\partial^2 \mathbf{v}}{\partial y^2} + \frac{\partial^2 \mathbf{v}}{\partial z^2} \quad . \quad (3.5)$$

Hier bezeichnet $p(\mathbf{x})$ den ortsabhängigen Druck und \mathbf{f} eine gerichtete, äußere Kraft. Das Gleichungssystem wird vervollständigt mit der Kontinuitätsgleichung

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\mathbf{v} \rho) = 0 \quad (3.6)$$

und der Gleichung zur Energieerhaltung

$$\frac{\partial \rho E}{\partial t} + \nabla \cdot (\mathbf{v}(\rho E + p)) = 0 \quad (3.7)$$

für die spezifische Energie E . Eine detaillierte Herleitung der Navier-Stokes-Gleichungen ist z.B. in Constantin und Foias, 1988 oder Temam, 2001 zu finden.

In dieser Arbeit werden ausschließlich eindimensionale Strömungen in x-Richtung betrachtet. Für diesen Fall reduziert sich das Gleichungssystem zu

$$\rho \left(\frac{\partial v_x}{\partial t} + v_x \frac{\partial v_x}{\partial x} \right) = F_x - \frac{\partial p}{\partial x} + \kappa \frac{\partial^2 v_x}{\partial x^2} + \frac{1}{3} \kappa \frac{\partial^2 v_x}{\partial x^2} \quad (3.8)$$

$$\frac{\partial \rho}{\partial t} + \frac{\partial v_x \rho}{\partial x} = 0 \quad (3.9)$$

$$\frac{\partial \rho E}{\partial t} + \frac{\partial v_x (\rho E + p)}{\partial x} = 0 \quad (3.10)$$

für die verbleibende Richtung x.

Die räumlich eindimensionale Darstellung in den Gleichungen Gl. (3.8) bis Gl. (3.10) ist der Ausgangspunkt für die weiteren Ausführungen zu den Euler-Gleichungen bzw. der Burgers-Gleichung. Die hier nicht adressierten mehrdimensionalen Fälle ergeben sich mit den entsprechenden Annahmen analog aus den Navier-Stokes-Gleichungen Gl. (3.2), Gl. (3.6) und Gl. (3.7).

3.1.2 Euler-Gleichungen

Die Euler-Gleichungen sind, ebenso wie die Navier-Stokes-Gleichungen, ein System aus drei partiellen Differentialgleichungen. Das Gleichungssystem ergibt sich aus Gl. (3.8 - 3.10) unter der Vernachlässigung von Reibungseinflüssen und Wärmeleitung innerhalb des Fluids. Damit reduziert sich die Gleichung für die Impulserhaltung in die ausschließlich betrachtete x-Richtung zu

$$\rho \left(\frac{\partial v_x}{\partial t} + v_x \frac{\partial v_x}{\partial x} \right) = F_x - \frac{\partial p}{\partial x} \quad . \quad (3.11)$$

Da für die hier betrachteten Anwendungen die Modellierung aufgeprägter Volumenkräften nicht notwendig ist, werden auch diese im Folgenden nicht weiter berücksichtigt. Es gilt $F_x = 0$. So lässt sich das Gleichungssystem aus Kontinuitätsgleichung, Impuls- und Energieerhaltung zu

$$\frac{\partial}{\partial t} \begin{bmatrix} \rho \\ \rho v_x \\ \rho E \end{bmatrix} + \frac{\partial}{\partial x} \begin{bmatrix} \rho v_x \\ \rho v_x^2 + p \\ v_x(\rho E + p) \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \quad (3.12)$$

vereinfachen.

3.1.3 Burgers-Gleichung

Eine weitere Vereinfachung der Navier-Stokes-Gleichungen ist die sogenannte Burgers-Gleichung. Es wird angenommen, dass keine Druckgradienten innerhalb des Fluids auftreten können und keine äußeren Kräfte auf das Strömungsfeld wirken. Im Gegensatz zu den Euler-Gleichungen wird allerdings die innere Reibung des Fluids berücksichtigt. Vor allem aufgrund des konstant angenommenen Drucks können reale Strömungen durch die Burgers-Gleichung nur sehr eingeschränkt modelliert werden. Die Impulserhaltung aus Gl. (3.8) reduziert sich folglich zu

$$\rho \left(\frac{\partial v_x}{\partial t} + v_x \frac{\partial v_x}{\partial x} \right) = \kappa \frac{\partial^2 v_x}{\partial x^2} \quad . \quad (3.13)$$

Mit der Einführung der kinematischen Viskosität $\nu = \kappa/\rho$ ergibt sich die sogenannte viskose Burgers-Gleichung

$$\frac{\partial v_x}{\partial t} + v_x \frac{\partial v_x}{\partial x} = \nu \frac{\partial^2 v_x}{\partial x^2} \quad . \quad (3.14)$$

Da Dichte und Druck durch die Vereinfachungen als konstant angenommen werden, ist das Strömungsfeld eindeutig über die Geschwindigkeit definiert. Daher sind neben der Impulserhaltung keine weiteren Differentialgleichungen notwendig, die Kopplungen zwischen verschiedenen Größen berücksichtigen.

Im Unterschied zu den Navier-Stokes- und den Euler-Gleichungen, existiert für die Burgers-Gleichung eine analytische Lösung. Diese kann beispielsweise über die Hopf-Coleman-Transformation (Gorguis, 2006) hergeleitet werden. Die Möglichkeit eines Abgleichs mit der analytischen Lösung hat die Burgers-Gleichung zu einem häufig genutzten Referenzbeispiel für numerische Lösungsverfahren in der Strömungssimulation gemacht. Obwohl in vielen Fällen Strömungen nicht zufriedenstellend mit der Burgers-Gleichung beschrieben werden können, wird sie aufgrund ihres Referenzcharakters und der einfachen Form auch in dieser Arbeit als Beispiel für den Umgang mit partiellen Differentialgleichungen in der Modellbildung genutzt.

3.1.4 Numerische Lösung

Eine allgemeine, geschlossene analytische Lösung existiert weder für die Navier-Stokes-Gleichungen noch für die Euler-Gleichung. Wie in Abschnitt 2.1 zum Stand der Forschung beschrieben, hat sich für die Lösung dieser Art von partiellen Differentialgleichungen vor allem die FVM etabliert. Bei diesem Ansatz wird zunächst die räumliche Dimension in Kontrollvolumina unterteilt. Innerhalb jeder Zelle wird der Zustand mit einem einzelnen Wert beschrieben, der sich meist als räumliche Mittelung über die Zelle ergibt. Für jedes Volumen werden Energie- und Stoffströme bilanziert. So ergibt sich eine gewöhnliche Differentialgleichung mit einer Dimension, die der Anzahl der räumlichen Diskretisierungsschritte entspricht. Für die Lösung der gewöhnlichen Differentialgleichung können dann gängige Zeitintegrationsverfahren, z.B. gemäß dem Runge-Kutta-Schema (Butcher, 1996), genutzt werden. Das grundlegende, mathematische Vorgehen bei der FVM ist in Anhang A erklärt.

Da für diese Arbeit mehrfach eine eindimensionale, numerische Strömungssimulation genutzt wird, die auf den Euler-Gleichungen basiert, wird hier kurz auf dieses spezielle Lösungsverfahren eingegangen. Zur Lösung des Euler-Gleichungssystems ist zunächst noch die Einführung einer algebraischen Zustandsgleichung notwendig, welche die einzelnen Zustandsgrößen in einen algebraischen Zusammenhang bringt. Dafür wird ein ideales Gas angenommen. Für den Zusammenhang zwischen Geschwindigkeit, Druck, Dichte und spezifischer Energie gilt damit zu jedem Zeitpunkt

$$E = \frac{1}{2} v_x^2 + \frac{p}{\rho(\gamma - 1)} \quad . \quad (3.15)$$

An dieser Stelle beschreibt γ den Isentropenexponenten.

Der für diese Arbeit verwendete Simulator wurde ebenfalls im Sonderforschungsbereich 1029 entwickelt mit dem Ziel, verschiedene Strömungsvorgänge in der beschriebenen Gasturbine mit Gleichraumverbrennung zu simulieren. Eine ausführliche Beschreibung der numerischen Umsetzung ist in Berndt, 2016 zu finden. Des Weiteren wurden mathematische Details in Berndt, 2014 und Berndt und Klein, 2016 veröffentlicht. Da die Simulation auch für Fluide aus mehreren Komponenten genutzt werden kann und entsprechende Reaktionen berücksichtigt, ist eine robuste, schwingungsfreie numerische Lösung äußerst komplex. Dies gilt insbesondere, da die Reaktionen möglicherweise aus Verbrennungsvorgängen resultieren, die mit starken Druck- und Temperaturschwankungen verbunden sind.

Das Gas wird als ideales Gas angenommen, da eine Modellierung als perfektes Gas mit entsprechend konstantem Isentropenexponenten γ dem, je nach Anwendung, möglicherweise großen Temperatur- und Druckbereich innerhalb einer Simulation nicht gerecht wird. Der Isentropenexponent $\gamma = \gamma(\vartheta)$ ist folglich abhängig von der Temperatur ϑ . Die Grundlage der numerischen Lösung ist die sogenannte Godunov-FVM (Harten, Lax & Van Leer, 1983; Einfeldt, 1988). Die Zustände in einem Volumen ergeben sich jeweils als arithmetisches Mittel über die entsprechende Zelle. Für die Berechnung der Ströme über die Grenzen der einzelnen Volumina hinweg ist die Lösung eines sogenannten Riemann-Problems an jedem Übergang notwendig. In der verwendeten Simulation werden die einzelnen Riemann-Probleme nicht exakt gelöst, da der Rechenaufwand dafür zu groß ist. Stattdessen wird eine approximative Lösung mit Hilfe einer Modifikation des Harten-Lax-Leer-Einfeldt-Lösers (Balsara, 2010) bestimmt, welche die entsprechenden Anforderungen an Robustheit und Schwingungsvermeidung aufweist. Die Berechnung der Strömung und der chemischen Kinetik erfolgt getrennt voneinander. Die Zeitschrittweite wird in jedem Schritt variabel so gewählt, dass die Courant–Friedrichs–Lewy-Bedingung, welche die numerische Stabilität garantiert (Giuliani

& Krivodonova, 2019), eingehalten wird. In jedem Zeitschritt wird zunächst ein halber Zeitschritt der chemischen Kinetik mit dem Löser RADAU15 (Hairer & Wanner, 1999) gerechnet, bevor der eigentliche strömungsmechanische Zeitschritt ausgewertet wird. Erst danach wird der fehlende halbe Zeitschritt der Kinetik ausgeführt. Für weitere Details zur verwendeten numerischen Simulation sei an dieser Stelle auf die ausführlichen Beschreibungen in Berndt, 2016 verwiesen.

Modellreduktion

Der Rechenaufwand der numerischen Lösung der Navier-Stokes- bzw. Euler-Gleichungen kann, je nach Wahl der räumlichen und zeitlichen Diskretisierung, stark variieren. Oftmals ist jedoch für eine zufriedenstellend genaue Abbildung von komplexen Strömungsphänomenen mit großen Gradienten in den Zustandsgrößen eine sehr feine räumliche und zeitliche Auflösung erforderlich, die folglich zu einem hohen Rechenaufwand führt. Mitunter wird für die Lösung selbst bei der Verwendung von Rechenclustern ein Vielfaches der simulierten Zeitspanne benötigt. Damit sind numerische Lösungen in ausreichender Qualität für Anwendungen, in denen die Rechenkapazität stark begrenzt ist, oft unbrauchbar. Dies gilt insbesondere, wenn die Auswertung des Modells in Echtzeit, z.B. für die Verwendung in einer Regelung, erfolgen soll.

Eine Möglichkeit, um den Rechenaufwand zu reduzieren, bieten Verfahren der Modellreduktion. Zwar geht eine Reduktion des Modells zumeist mit einem Verlust der Modellgenauigkeit einher, jedoch kann der mitunter größere Modellfehler in Anwendungen wie der Zustandsschätzung oder der Regelung kompensiert werden. Ein Beispiel für die Verwendung eines reduzierten Modells der Euler-Gleichungen im Kontext einer Zustandsschätzung ist in Schäpel et al., 2017 beschrieben. Allerdings ist auch in diesem Beispiel der Rechenaufwand nach wie vor deutlich zu hoch, um das reduzierte Modell in Echtzeit auszuwerten.

Da der nachfolgend beschriebene Ansatz der Approximation der Differentialgleichungslösung auf Basis von neuronalen Netzen eine Alternative zu den Möglichkeiten der Modellreduktion darstellen soll, werden hier stellvertretend drei verschiedene Verfahren der Modellreduktion kurz vorgestellt. Zudem wird eine mögliche Verwendung der reduzierten Modelle in einer Regelung oder Zustandsschätzung diskutiert. Dies soll die spätere Einordnung des neuen Ansatzes erleichtern. Ein ausführlicher Überblick über die verschiedenen, gängigen Methoden der Modellreduktion im Bereich der Fluidodynamik ist in Rowley und Dawson, 2017 zu finden.

- Die sogenannte POD ist die in der Strömungsmechanik am weitesten verbreitete Methode der Modellreduktion. Der Ansatz beruht auf der in der Statistik häufig verwendeten Hauptkomponentenanalyse. Ein Überblick über die Methode der POD ist in Berkooz, Holmes und Lumley, 1993 und Liang et al., 2002 gegeben. Die grundlegende Idee ist eine Darstellung des Modells in einem reduzierten Unterraum. Dafür wird zunächst eine Menge an räumlich und zeitlich aufgelösten Daten benötigt, welche als Schnappschüsse bezeichnet werden. Diese Daten werden üblicherweise aus einer Simulation mit dem vollen Modell erzeugt. Im Anschluss wird die Kovarianzmatrix der Schnappschüsse berechnet und deren Eigenwerte und Eigenvektoren bestimmt. Mit den Hauptkomponenten, die auch als Moden bezeichnet werden, kann ein Galerkin-Ansatz formuliert und auf das vollständige Modell projiziert werden. Es entsteht ein System aus gewöhnlichen, nichtlinearen Differentialgleichungen für die Amplituden der Moden. Abhängig von der Anzahl der berücksichtigten Hauptkomponenten ergibt sich ein mehr oder weniger stark reduziertes Modell. Dieses Verfahren stellt sicher, dass das reduzierte Modell

optimal bezüglich der totalen kinetischen Energie des Systems ist.

Da die POD in der Strömungsmechanik ein sehr populäres Verfahren zur Datenanalyse und Modellreduktion ist, existieren diverse Erweiterungen wie z.B. die *shifted*-POD-Methode (Reiss et al., 2018), die besonders für komplexe Transportphänome geeignet ist. Das Vorgehen bei der Erstellung von Schnappschüssen und die Bestimmung der Hauptkomponenten bleibt allerdings in allen Fällen gleich. Die Erstellung der Schnappschüsse ist der entscheidende Faktor für die zu erwartende Qualität des reduzierten Modells. Nur mit einer ausreichend großen Datenmenge, in der alle wichtigen Systemeigenschaften repräsentiert sind, kann ein reduziertes Modell gewonnen werden, das die relevanten Systemeigenschaften abbilden kann. Folglich bedeutet eine gute Approximation eines vorliegenden Datensatzes mit dem reduzierten Modell nicht zwangsläufig eine gute Abbildung der realen Systemdynamik. Zudem bildet das reduzierte Modell nur das Systemverhalten innerhalb eines fixen Zeithorizonts ab, in dem die Schnappschüsse erstellt wurden. Diese beiden Eigenschaften machen eine Verwendung in einer Regelungs- oder Zustandsschätzung problematisch. Vor allem die unbekannten Approximationsgenauigkeit bei Eingangsdaten, die in der Anpassung des Modells nicht verwendet wurden, kann in der Anwendung zu unvorhersehbarem Verhalten des Modells führen.

- *Balanced Truncation* ist wie die beschriebene POD eine Möglichkeit der Transformation des vollen Modells in einen passend gewählten Unterraum. Ein Überblick über die Methode ist in Gugercin und Antoulas, 2004 zu finden. Ausgangspunkt ist in diesem Fall eine Systemdarstellung mit einem Zustandsraummodell. Ziel ist es, ein reduziertes Modell zu finden, dass besonders gut steuer- und beobachtbar ist. In einem ersten Schritt werden die gramische Steuerbarkeits- und Beobachtbarkeitsmatrix für das volle Modell aufgestellt. Danach kann über eine Eigenwertzerlegung des Produktes der beiden Matrizen eine Transformation bestimmt werden, welche sicherstellt, dass das Produkt der transformierten gramischen Steuerbarkeits- und Beobachtbarkeitsmatrix eine Diagonalmatrix ergibt. Da die Lösung des letzten Schrittes nicht eindeutig ist, werden die transformierte Steuerbarkeits- und Beobachtbarkeitsmatrix anschließend noch *balanciert*. Das bedeutet, dass die Transformation so skaliert wird, dass die beiden Matrizen genau gleich sind. Aus der Transformation können dann gemäß der Eigenwerte die Richtungen ausgewählt werden, die in gleichem Maße am besten steuer- und beobachtbar sind, um ein reduziertes Modell aufzubauen.

Mit dem Fokus auf eine besonders gut steuer- und beobachtbare Transformation ist die *Balanced Truncation* explizit auf die Verwendung in einem Regelungs- bzw. Zustandsschätzverfahren ausgelegt. Das Verfahren stellt sicher, dass das dominierende Verhalten zwischen Ein- und Ausgängen des Systems erhalten bleibt. Trotzdem ergeben sich in der Anwendung mitunter einige Probleme. Falls das volle Modell aus einer physikalischen Beschreibung abgeleitet wird, sind die Zustände nach der Transformation nicht mehr physikalisch interpretierbar. Zudem kann nicht garantiert werden, dass durch die Transformation alle Systemeigenschaften, wie beispielsweise Stabilität, erhalten bleiben. Abschließend muss noch auf die Beschränkung der Methode auf lineare Zustandsraummodelle verwiesen werden. Damit ist eine Anwendung für viele reale Systeme vor allem im Bereich der Strömungsmechanik erst nach einer Linearisierung möglich, was mit einem beschränkten Anwendungsbereich des Modells einhergeht.

- Die *Port-Hamiltonische*-Modellierung ist ein relativ neuer Ansatz im Bereich der Modellreduktion. Ein Überblick über das Verfahren kann z.B. in van der Schaft, 2006 oder van der Schaft und Jeltsema, 2014 gefunden werden. Das Konzept sieht eine Untertei-

lung komplexer, multiphysikalischer Systeme in ein Netz aus einfachen hamiltonischen Blöcken vor, welche über sogenannte Ports verbunden sind. Über diese Ports können die Teilkomponenten Energie und Leistungsströme austauschen. Ein einzelnes Teilsystem besteht aus einem Speicherterm, einem Dissipationsterm sowie Pfaden für die Energieumformung und Verbindungs- bzw. Beschränkungsgleichungen. Mit diesem Aufbau kann sichergestellt werden, dass jedes Teilsystem bestimmte Eigenschaften hat, wie z.B. Passivität, Stabilität oder Energieerhaltung. Aus den Teilsystemen kann dann eine Beschreibung für das gesamte System aufgebaut werden.

Insgesamt sind viele Eigenschaften der *Port-Hamiltonischen*-Modellierung vielversprechend im Hinblick auf die Verwendung in einer Regelung oder Zustandsschätzung. Die Möglichkeit, das System beliebig zu erweitern, ohne die Eigenschaften der Teilsysteme zu beeinflussen, macht den Ansatz sehr gut skalierbar. Zudem verspricht die einfache Struktur eine recheneffiziente Auswertung ohne komplizierte, numerische Methoden. Allerdings ist die Umformung aus bekannten, mathematischen Beschreibungen in eine *Port-Hamiltonische*-Darstellung nicht trivial. Daher ist dieser Ansatz in vielen Bereichen bisher noch nicht so stark verbreitet. Erste Ansätze für den Bereich der Fluidodynamik sind in Rashad, Califano, Schuller und Stramigioli, 2021 und Bansal et al., 2021 zu finden.

3.2 Neuronale Netze

Wie bereits in Abschnitt 2.1 zum Stand der Technik kurz erwähnt, haben in den letzten Jahren vermehrt datengetriebene Ansätze aus dem Bereich des maschinellen Lernens den Weg in die Modellierung bzw. Regelung von Strömungssystemen gefunden. In dieser Arbeit werden ausschließlich die Möglichkeiten der Systemmodellierung bzw. der Approximation eines vorhandenen Modells mittels neuronaler Netze betrachtet.

Im Folgenden werden zunächst kurz die Grundlagen neuronaler Netze erläutert, um die Notation für die nächsten Abschnitte einzuführen. Da wie eingangs beschrieben, das Ziel ist, ein physikalisch motiviertes Zustandsraummodell für die Zeitreihenprädiktion zu erhalten, wird zudem kurz auf den gängigen Ansatz zur Zeitreihenprädiktion mithilfe neuronaler Netze eingegangen.

3.2.1 Netzarchitektur

Das grundlegende Element eines künstlichen neuronalen Netzes ist das einzelne Neuron. Der schematische Aufbau eines Neurons ist in Abb. 3.1 dargestellt. Das Neuron ist definiert über eine Eingangsfunktion α und eine Aktivierungsfunktion β . Die Eingangsfunktion α bildet dabei die durch den Vektor \mathbf{x} beschriebenen Eingangswerte zusammen mit einem Parametervektor \mathbf{w} und dem Bias b auf einen skalaren Wert v ab:

$$v = \alpha(\mathbf{x}, \mathbf{w}, b) \quad (3.16)$$

Nachgeschaltet wird der Ausgang des Neurons y über die Aktivierungsfunktion β berechnet:

$$y = \beta(v) \quad (3.17)$$

Für die Wahl der Eingangsfunktion α und der Aktivierungsfunktion β haben sich in der Anwendung bestimmte Konfigurationen als besonders praktikabel herausgestellt. In annähernd

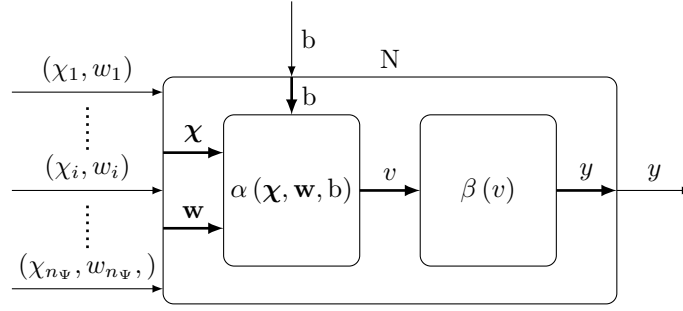


Abbildung 3.1: Schematische Darstellung eines allgemeinen Neurons N

allen Anwendungen wird für die Eingangsfunktion eine durch die Parameter \mathbf{w} gewichtete Summe der Eingangssignale gewählt:

$$v = \alpha(\boldsymbol{\chi}, \mathbf{w}, b) = \left[\sum_{i=1}^{n_\Psi} \chi_i \cdot w_i \right] + b = \boldsymbol{\chi}^T \mathbf{w} + b \quad (3.18)$$

Für die Aktivierungsfunktionen haben sich verschiedene Ansätze etabliert. Dabei spielt vor allem der erwartete Wertebereich der Eingänge eine Rolle. Auf den Wertebereich kann durch die Normierung der Trainingsdaten Einfluss genommen werden. Ein Überblick über die gängigsten Ansätze ist z.B. in Apicella, Donnarumma, Isgrò und Prevete, 2021 zu finden. Im Rahmen dieser Arbeit wurden zwei verschiedene Ansätze für die Aktivierungsfunktion verwendet. Für einfache Netzstrukturen kann eine hyperbolische Funktion

$$y = \beta(v) = \tanh(v) \quad (3.19)$$

gewählt werden. Eine Aktivierungsfunktion dieser Form hat den Vorteil, dass sich glatte Verläufe in den Ableitungen der Netzausgänge nach den jeweiligen Parametern bzw. Netzeingängen ergeben. Der Nachteil einer hyperbolischen Funktion ist, dass unstetiges Verhalten in den Trainingsdaten mitunter nur schlecht erlernt werden kann. Hinzu kommt, dass bei tiefen Netzstrukturen die Trainingssensitivität für die Parameter auf tiefen Schichten des Netzes stark abnimmt (Glorot & Bengio, 2010). Um diese Probleme zu umgehen, wird alternativ oft die sogenannte ReLu-Funktion

$$y = \beta(v) = \begin{cases} v & \text{für } v \geq 0 \\ 0 & \text{sonst} \end{cases} \quad (3.20)$$

verwendet. Je nach Wahl der Aktivierungsfunktion ist es notwendig, die Netzeingänge unterschiedlich zu normieren, um eine möglichst hohe Sensitivität im Trainingsprozess sicherzustellen. Für weitere Informationen zur Wahl der Aktivierungsfunktion sei auf die zuvor genannte Literatur verwiesen.

Einzelne Neuronen können beliebig verschaltet werden, um ein neuronales Netz aufzubauen. Dadurch ist es prinzipiell möglich, beliebig komplexe Netze aus Neuronen zusammenzustellen. In der Praxis haben sich allerdings bestimmte Strukturen etabliert. Da in dieser Arbeit vor allem die grundlegende Struktur der sogenannten „*Feed Forward*“-Netze verwendet wird, wird im Folgenden genauer auf diese Struktur eingegangen. Für weiterführende Informationen sei auf die bereits genannte Literatur verwiesen.

Werden die Neuronen in einzelnen Schichten gruppiert und Verbindungen innerhalb der Schicht ausgeschlossen, können die Eingänge, die Gewichte und die Biaswerte der j -ten

Schicht in den Tensoren χ_j , \mathbf{W}_j und \mathbf{b}_j zusammengefasst werden.¹ Sind die Eingänge der Neuronen in einer Schicht ausschließlich mit den Ausgängen der vorgeschalteten Schicht verbunden, wird dies als „*Feed Forward*“-Netz bezeichnet. Die schematische Netztopologie ist in Abb. 3.2 dargestellt. Mit der vektoriellen Aktivierungsfunktion β gilt dann für die Eingänge

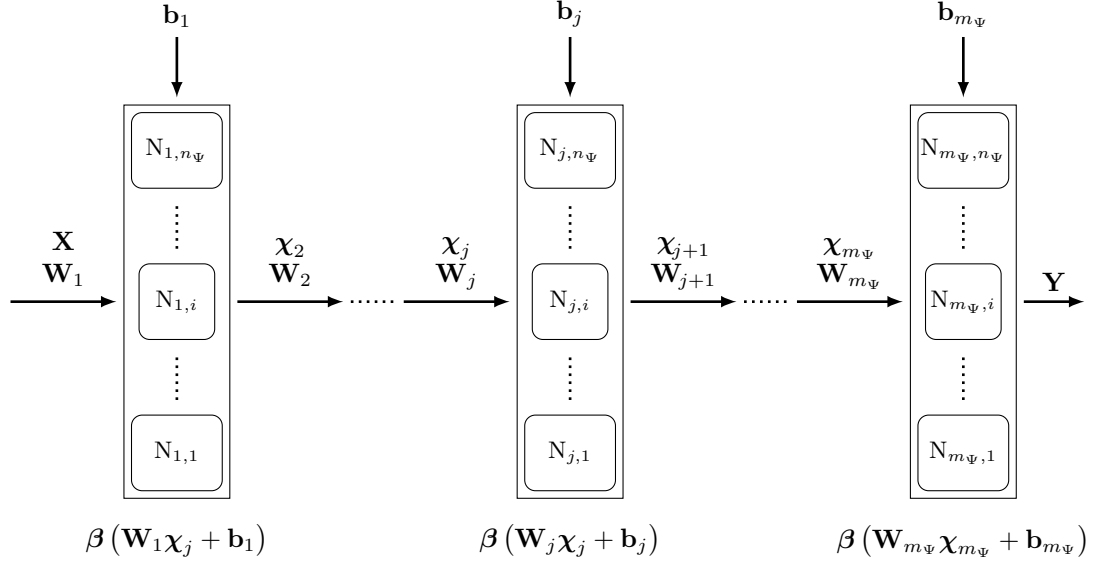


Abbildung 3.2: Topologie eines „*Feed Forward*“-Netzes

zweier hintereinander geschalteter Schichten

$$\chi_{j+1} = \beta(\mathbf{W}_j \chi_j + \mathbf{b}_j) \quad . \quad (3.21)$$

Ein „*Feed Forward*“-Netz $\Psi^{\mathcal{W}, \mathcal{B}}$ besteht aus m_Ψ Schichten und ist damit eindeutig durch die Menge aller Gewichte

$$\mathcal{W} = \{\mathbf{W}_1, \dots, \mathbf{W}_j, \dots, \mathbf{W}_{m_\Psi}\} \quad (3.22)$$

und aller Biaswerte

$$\mathcal{B} = \{\mathbf{b}_1, \dots, \mathbf{b}_j, \dots, \mathbf{b}_{m_\Psi}\} \quad (3.23)$$

definiert. Ein so definiertes „*Feed Forward*“-Netz $\Psi^{\mathcal{W}, \mathcal{B}}$ bildet folglich die Eingänge der ersten Schicht $\chi_1 = \mathbf{X}$ auf die Ausgänge \mathbf{Y} der letzten Schicht ab:

$$\mathbf{Y} = \Psi^{\mathcal{W}, \mathcal{B}}(\mathbf{X}) \quad (3.24)$$

3.2.2 Training

Um die gewünschte Abbildung von \mathbf{X} auf \mathbf{Y} zu erhalten, ist eine Anpassung der Gewichte \mathcal{W} und Biaswerte \mathcal{B} notwendig. Der Vorgang der Anpassung der Parameter wird im Allgemeinen als Training bezeichnet. Durch die weite Verbreitung von Methoden des maschinellen Lernens haben sich in den letzten Jahren viele verschiedene, spezialisierte Trainingsstrategien

¹Formal handelt es sich bei χ_j , \mathbf{W}_j und \mathbf{b}_j um Vektoren bzw. eine Matrix. Da für allgemeine Netzstrukturen in der Literatur allerdings häufig die Bezeichnung Tensor verwendet wird, wird sich hier an dieser Bezeichnung orientiert.

3 Modellierung

entwickelt. Da sich diese Arbeit nicht explizit mit den Trainingsmethoden neuronaler Netze beschäftigt, wird hier nur kurz auf das grundlegende Konzept des sogenannten überwachten Lernens neuronaler Netze eingegangen. Für detaillierte Informationen sei auf die Literatur (Larochelle, Bengio, Louradour & Lamblin, 2009; Zhang & Suganthan, 2016) verwiesen.

Für das überwachte Lernen werden Trainingsdaten benötigt, die Beispiele für die gewünschte Eingangs-/Ausgangsabbildung des Netzes darstellen. In der Literatur wird hier von gelabelten Daten gesprochen. Ein Trainingspaar, häufig auch als Sample bezeichnet, besteht aus einem Eingangstensor $\mathbf{X}_{T,i}$ und dem gewünschten Ausgangstensor $\mathbf{Y}_{T,i}$, der als Label bezeichnet wird. Die Menge aller n_T Ein- und Ausgangsdatenpaare, die im Training berücksichtigt werden sollen, kann zusammengefasst werden in

$$\mathcal{X}_T = \{\mathbf{X}_{T,1}, \dots, \mathbf{X}_{T,i}, \dots, \mathbf{X}_{T,n_T}\} \quad (3.25)$$

und

$$\mathcal{Y}_T = \{\mathbf{Y}_{T,1}, \dots, \mathbf{Y}_{T,i}, \dots, \mathbf{Y}_{T,n_T}\} \quad . \quad (3.26)$$

Für die Eingangsdaten \mathcal{X}_T kann mit dem Netz $\Psi^{\mathcal{W},\mathcal{B}}$ eine Menge an Ausgangsdaten

$$\mathcal{Y}_P = \{\mathbf{Y}_{P,1}, \dots, \mathbf{Y}_{P,i}, \dots, \mathbf{Y}_{P,n_T}\} \quad \text{mit} \quad \mathbf{Y}_{P,i} = \Psi^{\mathcal{W},\mathcal{B}}(\mathbf{X}_{T,i}) \quad \mathbf{X}_{T,i} \in \mathcal{X}_T \quad (3.27)$$

prädiziert werden. Die prädizierten Ausgänge \mathcal{Y}_P können dann mit den gewünschten Trainingsausgängen \mathcal{Y}_T verglichen werden, um mithilfe einer Trainingsfunktion L einen skalaren Fehler

$$e_L = L(\mathcal{Y}_P, \mathcal{Y}_T) \quad (3.28)$$

zu bestimmen. Der Fehler e_L ist das Maß für die Anpassung an die Trainingsdaten in Abhängigkeit der Parameter des Netzes, das heißt der Gewichte \mathcal{W} und Biaswerte \mathcal{B} :

$$L(\mathcal{Y}_P(\mathcal{W}, \mathcal{B}, \mathcal{X}_T), \mathcal{Y}_T) = L(\mathcal{W}, \mathcal{B}, \mathcal{X}_T, \mathcal{Y}_T) \quad . \quad (3.29)$$

Folglich können die optimalen Parameter aus einer Optimierung

$$\mathcal{W}, \mathcal{B} = \arg \min_{\mathcal{W}, \mathcal{B}} L(\mathcal{W}, \mathcal{B}, \mathcal{X}_T, \mathcal{Y}_T) \quad (3.30)$$

ermittelt werden. Für die Lösung des Optimierungsproblems in Gl. (3.30) werden vor allem Verfahren verwendet, die das Minimum über einen Gradientenabstieg bestimmen. Dafür wird der Gradient der Trainingsfunktion ∇L in Abhängigkeit der Parameter \mathcal{W}, \mathcal{B} bestimmt und die Parameter entsprechend angepasst. Ein einzelner Schritt der Optimierung wird als Epoche bezeichnet. Da sich damit für diese Arbeit besonders zuverlässige Trainingsergebnisse gezeigt haben, wird in dieser Arbeit abweichend zur gängigen Praxis für das Training der neuronalen Netze der Broyden–Fletcher–Goldfarb–Shanno (BFGS)-Algorithmus (Fei, Rong, Wang & Wang, 2014) benutzt. Dieses Quasi-Newton-Verfahren mit einer iterativen Schätzung der Hesse-Matrix beschleunigt das Lernverhalten für die speziellen Probleme dieser Arbeit erheblich. Der grundlegende BFGS-Algorithmus ist in Anhang B erklärt. In den meisten Fällen außerhalb dieser Arbeit gleicht allerdings der Aufwand zur Schätzung der Hesse-Matrix den Trainingsvorteil nicht aus, weshalb dort auf die Berücksichtigung der Hesse-Matrix im Optimierungsverfahren verzichtet wird.

3.2.3 Zeitreihenprädiktion

Da der unten beschriebene Modellierungsansatz auf Basis von PINNs zur Zeitreihenprädiktion verwendet wird, soll an dieser Stelle kurz auf das verbreitetste Verfahren zur Zeitreihenprädiktion auf Basis neuronaler Netze eingegangen werden. Damit soll die spätere Einordnung des PINN-Ansatzes erleichtert werden.

In den letzten Jahren haben sich aufgrund vieler erfolgreicher Anwendungen sogenannte *Long Short-Term Memory* (LSTM)-Netze für die Zeitreihenprädiktion etabliert. Ein Überblick über die Theorie von LSTM-Netzen ist z.B. in Sherstinsky, 2020 oder Lindemann et al., 2021 zu finden. Ziel der Verwendung von LSTM-Netzen ist das Erlernen eines zeitlichen Zusammenhangs in den Eingangsdaten. Folglich spielt es im Gegensatz zum Trainingsvorgang in einem konventionellen neuronalen Netz eine Rolle, in welcher Reihenfolge die Eingangsdaten in das Netz gegeben werden. Der strukturelle Unterschied sind die Rückführung vergangener Ausgänge eines Neurons sowie die Berücksichtigung eines inneren Zellzustandes \mathbf{c}_l , der über die Zeitschritte gespeichert wird.

Der Aufbau einer LSTM-Schicht ist in Abb. 3.3 dargestellt. Am blau eingefärbten Multiplexer wird der Netzeingang $\mathbf{x}_{j,l}$ mit dem Netzausgang $\mathbf{y}_{j,l-1}$ aus dem vorangegangenen Zeitschritt zusammengeführt zu

$$\mathbf{U} = \begin{bmatrix} \mathbf{x}_{j,l} \\ \mathbf{y}_{j,l-1} \end{bmatrix} . \quad (3.31)$$

Die gelb gekennzeichneten Blöcke entsprechen den grundlegenden Schichten eines neuronalen Netzes, wie es in Abb. 3.2 dargestellt ist. Zu jedem dieser Blöcke gehören separat trainierbare Gewichtsmatrizen und Biaswerte, so dass gilt

$$\sigma_v(\mathbf{U}) = \sigma(\mathbf{W}_v \mathbf{U} + \mathbf{b}_v) \quad (3.32)$$

$$\mathbf{tanh}_{it}(\mathbf{U}) = \mathbf{tanh}(\mathbf{W}_{it} \mathbf{U} + \mathbf{b}_{it}) \quad (3.33)$$

$$\sigma_i(\mathbf{U}) = \sigma(\mathbf{W}_i \mathbf{U} + \mathbf{b}_i) \quad (3.34)$$

$$\sigma_o(\mathbf{U}) = \sigma(\mathbf{W}_o \mathbf{U} + \mathbf{b}_o) . \quad (3.35)$$

Hier beschreiben $\sigma(\cdot)$ und $\mathbf{tanh}(\cdot)$ die sigmoide bzw. tangens-hyperbolische, elementweise Aktivierungsfunktion. Die Ausgänge der einzelnen Funktionen werden über elementweise Operationen (in Abb. 3.3 orange, rot und grün gekennzeichnet) zusammengeführt, um die Ausgänge \mathbf{y} und \mathbf{c} zu bestimmen. Wie durch die gestrichelte Linie in Abb. 3.3 gekennzeichnet, werden die beiden Werte über die Zeitschritte hinweg gespeichert. Der Ausgang der LSTM-Schicht kann dann entsprechend auf eine weitere LSTM-Schicht oder eine beliebige andere Schicht weitergeleitet werden. In den meisten Anwendungen wird mindestens auf der letzten Schicht des Netzes eine konventionelle „Feed Forward“-Schicht verwendet, um die Daten entsprechend dem gewünschten Ausgangsformat zu bündeln.

Die beschriebene Struktur einer LSTM-Einheit ermöglicht es, im Training die Parameter des Netzes so anzupassen, dass die für eine gute Prädiktion relevanten Daten aus vergangenen Eingängen herausgefiltert werden. Trotz dieser Fähigkeit hat die Wahl der Eingangsdaten einen gravierenden Einfluss auf die Qualität der mit einem LSTM-Netz erzielten Prädiktion. Wie bei klassischen Verfahren der Zeitreihenidentifikation (siehe Abschnitt 2.1) ist auch hier zu entscheiden, wie das Zeitfenster der Eingangsdaten zu wählen ist und ob sowohl vergangene Eingänge als auch Ausgänge verwendet werden sollen.

Wie bei allen datengetriebenen Modellen ist auch die Qualität einer Prädiktion mit einem LSTM-Netz stark abhängig von den zur Verfügung stehenden Daten. Gleiches gilt für die

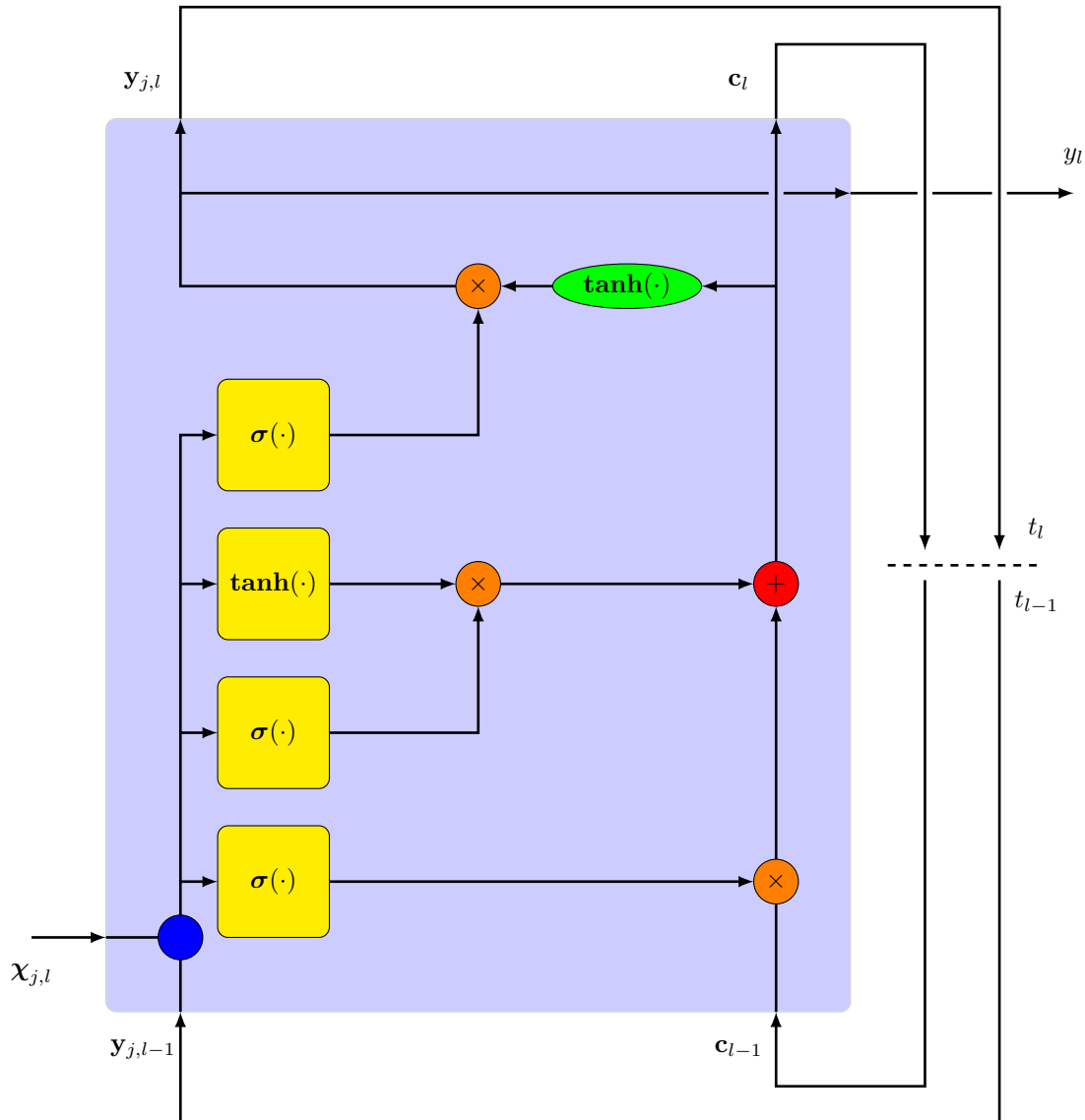


Abbildung 3.3: Eine Schicht/Einheit eines LSTM-Netzes

Robustheit und Generalisierbarkeit. Das physikalische Verhalten des Systems wird hier nicht berücksichtigt. Damit ist die Verwendung von LSTM-Netzen insbesondere schwierig, wenn die Datenmenge im Training stark begrenzt ist. Dies limitiert die Anwendung vor allem in Fällen, in denen Ein- und Ausgangsdaten über aufwendige Versuche erzeugt werden müssen. Desweiteren sollte angemerkt werden, dass der Rechenaufwand für LSTM-Netze durch die kompliziertere Struktur sowohl im Trainingsprozess als auch in der Inferenz gegenüber einfachen „Feed Forward“-Netzen deutlich ansteigt.

3.3 Physics-Informed Neural Networks

Wie die Diskussion im vorangegangenen Abschnitt zeigt, ist die Qualität der Zeitreihenprädiktion mit neuronalen Netzen immer direkt abhängig von Wahl und Umfang der Trainingsdaten. Systemdynamische Informationen, die nicht in den Trainingsdaten repräsentiert sind, werden nicht in der Systembeschreibung mit dem trainierten neuronalen Netz enthal-

ten sein. Dadurch führen Approximationen dieser Art für Eingangsdaten, die nicht in der Anpassung des Modells verwendet wurden, häufig zu ungenaueren Ergebnissen als physikalische Beschreibungen mit partiellen Differentialgleichungen. Eine Möglichkeit, physikalische Information über das System in die Beschreibung mit einem neuronalen Netz einzubeziehen, ist die Verwendung von PINNs, dessen Konzept in Raissi, 2018 vorgestellt wurde. Im Unterschied zu den konventionellen Ansätzen zur Zeitreihenapproximation wird in diesem Ansatz nicht mithilfe eines neuronalen Netzes nur das Eingangs-/Ausgangsverhalten einer Zeitreihe prädiziert, stattdessen wird das neuronale Netz trainiert, die Lösung einer partiellen Differentialgleichung zu approximieren. Teile der folgenden Ausführungen wurden bereits in Arnold und King, 2021b veröffentlicht.

Ausgangspunkt ist eine allgemeine partielle Differentialgleichung in der Form

$$\frac{\partial h}{\partial t} + \mathcal{N}(h) = 0 \quad (3.36)$$

mit der Lösung $h(\mathbf{x}, t)$. Für die hier verwendete Notation wird sich an den Ausführungen in Raissi, 2018 orientiert. Im Folgenden bezeichnet \mathcal{N} einen beliebigen Differentialoperator. Da in dieser Arbeit nur räumlich eindimensionale Probleme betrachtet werden, ist die Lösung der partiellen Differentialgleichung am Punkt (\mathbf{x}, t) in der Raum-Zeit-Ebene durch $h(\mathbf{x}, t)$ gegeben. Die Lösung der partiellen Differentialgleichung soll nun mit einem einfachen „Feed Forward“-Netz $\Psi^{\mathcal{W}, \mathcal{B}}$, wie es in Abschnitt 3.2 beschrieben wurde, angenähert werden:

$$h(\mathbf{x}, t) \approx \Psi^{\mathcal{W}, \mathcal{B}}(\mathbf{X}) \quad \text{mit} \quad \mathbf{X} = [\mathbf{x}, t]^T \quad (3.37)$$

Entsprechend sollten die partiellen Ableitungen des neuronalen Netzes ebenfalls die Differentialgleichung über die Funktion

$$\theta := \frac{\partial \Psi^{\mathcal{W}, \mathcal{B}}}{\partial t} + \mathcal{N}(\Psi^{\mathcal{W}, \mathcal{B}}) \stackrel{!}{=} 0 \quad (3.38)$$

möglichst gut erfüllen. Die Ableitungen können an dieser Stelle über das Verfahren des automatischen Differenzierens bestimmt werden. Ein Überblick über die Methodik ist in Baydin, Pearlmutter, Radul und Siskind, 2018 gegeben. Die linke Seite von Gl. (3.38) sowie die Approximation der Lösung der partiellen Differentialgleichung in Gl. (3.37) teilen die Parameter \mathcal{W}, \mathcal{B} und können daher beide für ein Training berücksichtigt werden.

Die Trainingsfunktion für die Anpassung der Parameter \mathcal{W}, \mathcal{B} setzt sich folglich aus zwei Teilfunktionen zusammen. Dafür wird angenommen, dass die Lösung der Differentialgleichung $h(\mathbf{x}_{i_h}, t_{i_h})$ an den Stellen $(\mathbf{x}_{i_h}, t_{i_h})$ mit $i_h \in \{1, \dots, n_{Th}\}$ bekannt ist. Diese Punkte werden in der Trainingsmenge

$$\mathcal{X}_{T,h} = \{\mathbf{X}_{T,h}^1, \dots, \mathbf{X}_{T,h}^{i_h}, \dots, \mathbf{X}_{T,h}^{n_{Th}}\} \quad \text{mit} \quad \mathbf{X}_{T,h}^{i_h} = [\mathbf{x}_{i_h}, t_{i_h}]^T \quad (3.39)$$

$$\mathcal{Y}_{T,h} = \{\mathbf{Y}_{T,h}^1, \dots, \mathbf{Y}_{T,h}^{i_h}, \dots, \mathbf{Y}_{T,h}^{n_{Th}}\} \quad \text{mit} \quad \mathbf{Y}_{T,h}^{i_h} = h(\mathbf{x}_{i_h}, t_{i_h}) \quad (3.40)$$

zusammengefasst. Üblicherweise ist die Lösung der Differentialgleichung mindestens an den Rändern der Raum-Zeit-Ebene über gegebene Anfangs- und Randbedingungen bekannt. Allerdings können in den Trainingsdaten auch zusätzliche Informationen z.B. aus Messungen oder numerischen Simulationen berücksichtigt werden. Letzteres wird weiter unten für die Annäherung der Euler-Gleichungen Verwendung finden. Für diese Trainingsdaten kann mit dem mittleren quadratischen Fehler eine Trainingsfunktion

$$L_h(\mathcal{W}, \mathcal{B}, \mathcal{X}_{T,h}, \mathcal{Y}_{T,h}) = \frac{1}{n_{Th}} \sum_{i_h=1}^{n_{Th}} \left(\mathbf{Y}_{T,h}^{i_h} - \mathbf{Y}_{P,h}^{i_h} \right)^2 \quad \text{mit} \quad \mathbf{Y}_{P,h}^{i_h} = \Psi^{\mathcal{W}, \mathcal{B}}(\mathbf{X}_{T,h}^{i_h}) \quad (3.41)$$

3 Modellierung

definiert werden.

In der zweiten Teilfunktion des Trainings wird das PINN aus Gl. (3.38) berücksichtigt. Dafür wird eine separate Menge an Trainingsdaten

$$\mathcal{X}_{T,\theta} = \{\mathbf{X}_{T,\theta}^1, \dots, \mathbf{X}_{T,\theta}^{i_\theta}, \dots, \mathbf{X}_{T,\theta}^{n_{T,\theta}}\} \quad \text{mit} \quad \mathbf{X}_{T,\theta}^{i_\theta} = [x_{i_\theta}, t_{i_\theta}]^T \quad (3.42)$$

aus $n_{T,\theta}$ Kollokationspunkten $(x_{i_\theta}, t_{i_\theta})$ definiert, an denen die durch θ festgelegte partielle Differentialgleichung erfüllt sein soll. Damit lässt sich die zweite Trainingsfunktion

$$L_\theta(\mathcal{W}, \mathcal{B}, \mathcal{X}_{T,\theta}) = \frac{1}{n_{T,\theta}} \sum_{i_\theta=1}^{n_{T,\theta}} \theta_{i_\theta} \quad \text{mit} \quad \theta_{i_\theta} = \frac{\partial \Psi^{\mathcal{W}, \mathcal{B}}}{\partial t}(\mathbf{X}_{T,\theta}^{i_\theta}) + \mathcal{N}(\Psi^{\mathcal{W}, \mathcal{B}}(\mathbf{X}_{T,\theta}^{i_\theta})) \quad (3.43)$$

aufstellen. Durch die Forderung für die Ableitungen in Gl. (3.38), die sich aus der partiellen Differentialgleichung ergibt, ist eine Festlegung von gewünschten Netzausgängen $\mathcal{Y}_{T,\theta}$ weder notwendig noch möglich. Die gewünschten Ausgänge ergeben sich lediglich aus dem durch die partielle Differentialgleichung definierten Zusammenhang.

Die beiden Trainingsfunktionen L_h und L_θ können für das Training nun zusammengefasst werden:

$$L(\mathcal{W}, \mathcal{B}, \mathcal{X}_{T,h}, \mathcal{Y}_{T,h}, \mathcal{X}_{T,\theta},) = L_h(\mathcal{W}, \mathcal{B}, \mathcal{X}_{T,h}, \mathcal{Y}_{T,h}) + L_\theta(\mathcal{W}, \mathcal{B}, \mathcal{X}_{T,\theta}) \quad . \quad (3.44)$$

Durch Verwendung der kombinierten Trainingsfunktion L mit den entsprechenden Trainingsdaten, kann für die Punkte, an denen die Lösung bekannt ist, eine gute Anpassung erzielt werden und zusätzlich an beliebig vielen Kollokationspunkten eine Erfüllung der partiellen Differentialgleichung gefordert werden.

Beispiel Burgers-Gleichung

Das Vorgehen für die Anpassung eines PINNs soll im Folgenden anhand der viskosen Burgers-Gleichung gemäß Gl. (3.14) verdeutlicht werden. Es wird sich ebenfalls an den Ausführungen in Raissi, 2018 orientiert. Gesucht wird ein PINN, das die Lösung der viskosen Burgers-Gleichung für bekannte Anfangs- und Randbedingungen

$$v_x(x, 0) = -\sin(\pi x) \quad (3.45)$$

$$v_x(-1, t) = v_x(1, t) = 0 \quad (3.46)$$

im Bereich $x \in [-1, 1]$ und für die Zeit $t \in [0, 1]$ s möglichst gut annähert. Die Viskosität wird hier mit einem Wert von $\nu = 0.2$ angenommen.

Für den ersten Teil der Trainingsfunktion werden zunächst $n_{T,h}$ Punkte (x_{i_h}, t_{i_h}) festgelegt, an denen die Lösung der Differentialgleichung bekannt ist. In diesem Fall wird angenommen, dass nur Anfangs- und Randbedingungen gemäß Gl. (3.45) bzw. Gl. (3.46) bekannt sind. Sie werden mit insgesamt $n_{T,h} = 1203$ Punkten in den Eingangstrainingsdaten $\mathcal{X}_{T,h}$ und entsprechenden Labels $\mathcal{Y}_{T,h}$ berücksichtigt. Die Daten bestehen dabei aus 201 äquidistanten Punkten zwischen $x = -1$ und $x = 1$ für die Anfangsbedingung und jeweils 501 äquidistanten Punkten zwischen $t = 0$ s und $t = 1$ s für die Randbedingungen. Um die Differentialgleichung durch das Netz anzunähern, muss zusätzlich

$$\theta := \frac{\partial \Psi^{\mathcal{W}, \mathcal{B}}}{\partial t} + \Psi^{\mathcal{W}, \mathcal{B}} \frac{\partial \Psi^{\mathcal{W}, \mathcal{B}}}{\partial x} - \nu \frac{\partial^2 \Psi^{\mathcal{W}, \mathcal{B}}}{\partial^2 x} \stackrel{!}{=} 0 \quad (3.47)$$

gelten. Diese Anforderung wird durch insgesamt $n_{T\theta} = 6203$ Punkte $(x_{i\theta}, t_{i\theta})$ in der Raum-Zeit-Ebene im Training berücksichtigt. Dabei werden 5000 zufällig gleichverteilte Kollokationspunkte im Inneren der Raum-Zeit Ebene ausgewählt und durch die 1203 Randpunkte aus der ersten Trainingsmenge ergänzt.

Für die Netzarchitektur wurde ein „Feed Forward“-Netz mit acht verborgenen Schichten aus jeweils 20 Neuronen gewählt. Dies entspricht der Struktur, die in Raissi, 2018 als bester Kompromiss zwischen Trainingsaufwand und Genauigkeit gewählt wurde. Die Eingangsschicht besteht aus zwei Neuronen entsprechend den Parametern der partiellen Differentialgleichung x und t . Die Ausgangsschicht gibt mit einem Neuron den skalaren Ausgang aus, der im Fall der Burgers-Gleichung der Geschwindigkeit v_x entspricht. Das Training wird mit dem BFGS-Algorithmus durchgeführt. Die Anzahl der Trainingsepochen wird nicht fest gewählt. Stattdessen wird eine minimale Änderung der Trainingsfunktion von 10^{-6} zwischen zwei Epochen festgelegt.

In Abb. 3.4 oben ist ein Intensitätsplot der Netzauswertung auf einem Raum-Zeit-Gitter mit 500 äquidistanten Punkten in der Zeitdimension und 200 äquidistanten Punkten in der räumlichen Dimension dargestellt. Zusätzlich ist der Fehler gegenüber einer numerischen Lösung

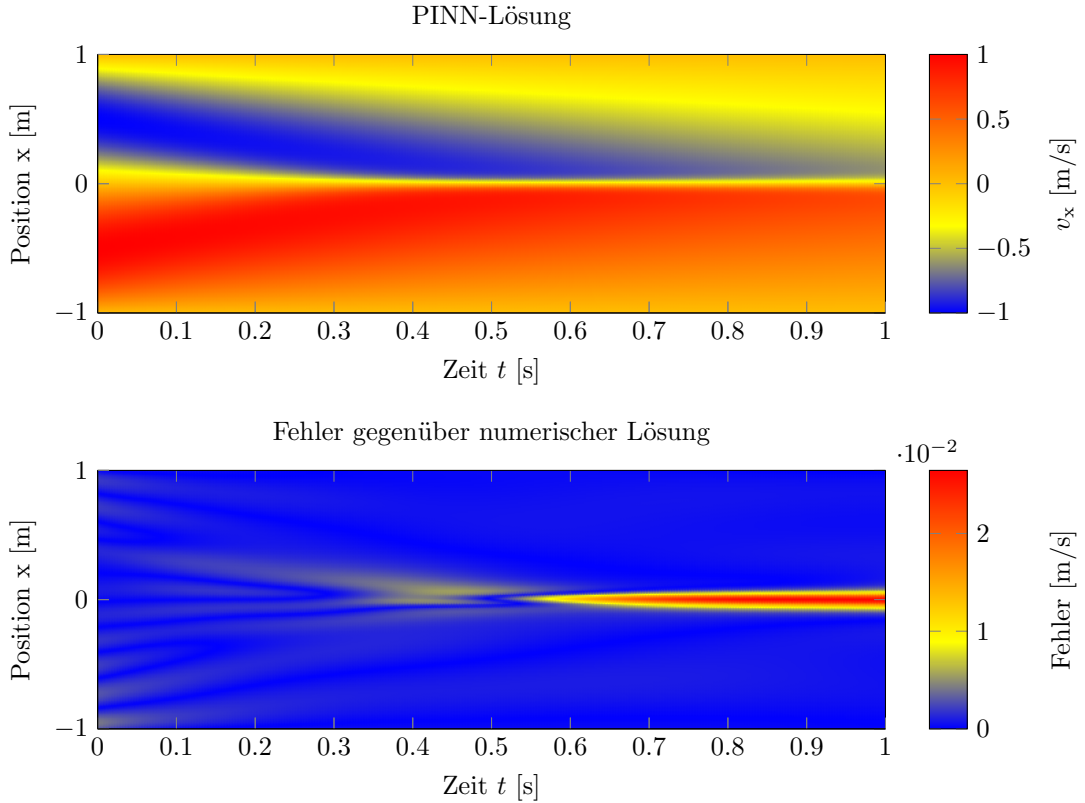


Abbildung 3.4: PINN-Lösung der Burgers-Gleichung aus Training mit bekannter Anfangsbedingung.

Oben: Intensitätsplot in der Raum-Zeit-Ebene

Unten: Fehler gegenüber einer numerischen Lösung mit einer FVM

auf dem gleichen Raum-Zeit-Gitter mit einer FVM abgebildet.² Die geringe Abweichung

²Wie beschrieben wäre an dieser Stelle auch ein Vergleich mit einer analytischen Lösung möglich gewesen. Die numerische Lösung wurde gewählt, da im weiteren Verlauf dieses Kapitels eine Interpolation aus teilweise zufälligen Werten zur Beschreibung der Anfangsbedingung verwendet wird und eine Berechnung der analytischen Lösung damit sehr aufwendig ist.

3 Modellierung

in der Größenordnung von 10^{-2} zeigt, dass die Lösung der partiellen Differentialgleichung durch das trainierte Netz ausgesprochen gut approximiert werden kann. Dies zeigt sich auch in Abb. 3.5, in der die Lösung zu einzelnen Zeitpunkten über der räumlichen Koordinate x dargestellt ist. Die PINN-Lösung und die numerische Lösung sind hier kaum voneinander zu

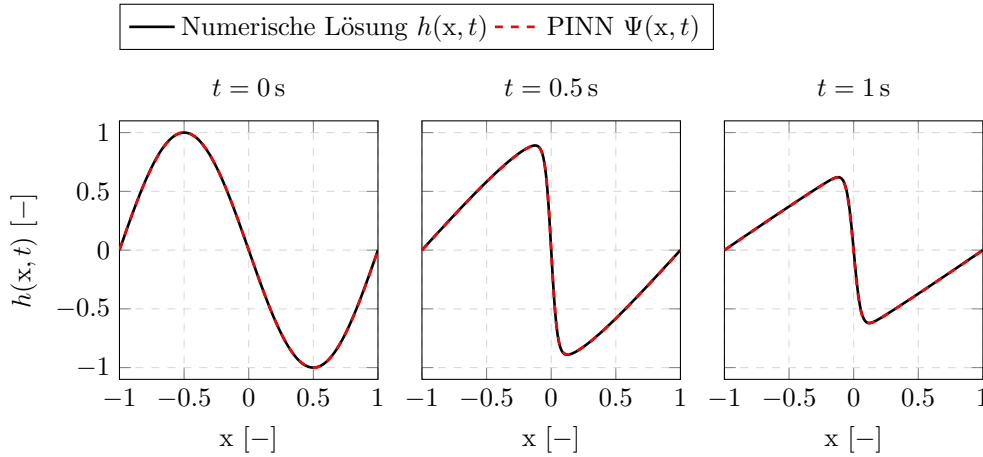


Abbildung 3.5: Vergleich zwischen PINN-Lösung aus einem Training mit bekannter Anfangsbedingung und einer numerischen Lösung mit einer FVM zu den Zeitpunkten $t = 0$ s, $t = 0.5$ s und $t = 1.0$ s.

unterscheiden.

3.4 Zustandsraummodellierung auf Basis von Physics-Informed-Neural-Networks

Die bisherigen Ausführungen, welche grundsätzlich die Arbeiten in Raissi, 2018 und Raissi, Perdikaris und Karniadakis, 2019 widerspiegeln, beziehen sich ausschließlich auf autonome partielle Differentialgleichungen und sind damit nicht direkt in den gängigen Ansätzen der modellgestützten Regelung oder Zustandsschätzung verwendbar. Wünschenswert für die genannten Anwendungen ist die Beschreibung der Systemdynamik zwischen der Aktuierung³ μ und der Messgröße γ beispielsweise in einer diskreten Zustandsraumdarstellung

$$\mathbf{x}_{l+1} = \mathbf{f}(\mathbf{x}_l, \mu_l) \quad (3.48)$$

$$\gamma_l = \mathbf{z}(\mathbf{x}_l, \mu_l) \quad . \quad (3.49)$$

Hier beschreiben \mathbf{f} bzw. \mathbf{z} die Zustands- bzw. Messfunktion für den Zeitschritt l . Der im vorangegangenen Abschnitt gezeigte Ansatz gibt eine Lösung in einem festen Zeitintervall $t \in [0, T]$ mit bekannten Anfangs- und Randbedingungen für ein autonomes System $\mu = 0$. Im Gegensatz dazu beschreiben Gl. (3.48) und Gl. (3.49) die Systemdynamik eines aktuierten Systems für einen beliebigen Zeitschritt l und damit auch auf einem beliebigen Zeithorizont. Der Systemzustand kann zum Zeitpunkt $t = 0$ mit einer bekannten Funktion

$$h(\mathbf{x}, 0) \quad (3.50)$$

beschrieben werden. Für die Darstellung in einem Zustandsraum wie in Gl. (3.48) und Gl. (3.49) wird allerdings eine Zustandsbeschreibung mit einer endlichen Anzahl an Zuständen im Zustandsvektor \mathbf{x} vorausgesetzt. Um einen Zustand mit einer endlichen Anzahl an

³Die sonst übliche Bezeichnung mit u wird unten für eine ganzzahlige Stellgröße verwendet

3.4 Zustandsraummodellierung auf Basis von Physics-Informed-Neural-Networks

Komponenten zu erhalten, kann die räumliche Verteilung von h zum Zeitpunkt $t = 0$, sprich die Anfangsbedingung, an n_ω Stützpunkten $\omega = [x_1, \dots, x_i, \dots, x_{n_\omega}]^T$ verwendet werden. Der diskrete Zustand zum Zeitpunkt $t = 0$ ist dann über

$$\mathbf{x}_0 = [h(x_1, 0), \dots, h(x_i, 0), \dots, h(x_{n_\omega}, 0)]^T \quad (3.51)$$

definiert. Aus den Stützstellen ω und dem darüber definierten Zustand \mathbf{x}_0 , kann über eine Interpolation ξ wiederum die ursprüngliche Anfangsbedingung approximiert werden:

$$h(x, 0) \approx \xi(x, \mathbf{x}_0, \omega) \quad . \quad (3.52)$$

Mit dieser Näherung für die Anfangsbedingung können nun die entsprechenden Punkte in der Trainingsfunktion Gl. (3.41) durch die Interpolation berücksichtigt werden. Um im Folgenden die Unterschiede zu verdeutlichen, werden diese Punkte in einer separaten Trainingsmenge

$$\mathcal{X}_{T,h_0} = \{\mathbf{X}_{T,h_0}^1, \dots, \mathbf{X}_{T,h_0}^{i_h}, \dots, \mathbf{X}_{T,h_0}^{n_{T,h_0}}\} \quad \text{mit} \quad \mathbf{X}_{T,h_0}^{i_h} = [x_{i_h}, 0]^T \quad (3.53)$$

$$\mathcal{Y}_{T,h_0} = \{\mathbf{Y}_{T,h_0}^1, \dots, \mathbf{Y}_{T,h_0}^{i_h}, \dots, \mathbf{Y}_{T,h_0}^{n_{T,h_0}}\} \quad \text{mit} \quad \mathbf{Y}_{T,h_0}^{i_h} = \xi(x_{i_h}, \mathbf{x}_0, \omega) \quad (3.54)$$

berücksichtigt. Für diese Trainingsmenge wird entsprechend eine Trainingsfunktion

$$L_{h_0}(\mathcal{W}, \mathcal{B}, \mathcal{X}_{T,h_0}, \mathcal{Y}_{T,h_0}) = \frac{1}{n_{T,h_0}} \sum_{i_h=1}^{n_{T,h_0}} \left(\mathbf{Y}_{T,h_0}^{i_h} - \mathbf{Y}_{P,h_0}^{i_h} \right)^2 \quad \text{mit} \quad \mathbf{Y}_{P,h_0}^{i_h} = \Psi^{\mathcal{W}, \mathcal{B}}(\mathbf{X}_{T,h_0}^{i_h}) \quad (3.55)$$

definiert. Damit erweitert sich die gesamte Trainingsfunktion zu

$$\begin{aligned} L(\mathcal{W}, \mathcal{B}, \mathcal{X}_{T,h}, \mathcal{Y}_{T,h}, \mathcal{X}_{T,\theta}, \mathcal{X}_{T,h_0}, \mathcal{Y}_{T,h_0}) &= L_h(\mathcal{W}, \mathcal{B}, \mathcal{X}_{T,h}, \mathcal{Y}_{T,h}) \\ &+ L_\theta(\mathcal{W}, \mathcal{B}, \mathcal{X}_{T,\theta}) \\ &+ L_{h_0}(\mathcal{W}, \mathcal{B}, \mathcal{X}_{T,h_0}, \mathcal{Y}_{T,h_0}) \quad . \end{aligned} \quad (3.56)$$

und es kann ein PINN $\Psi_0^{\mathcal{W}_0, \mathcal{B}_0}$ trainiert werden, welches eine durch den Zustand \mathbf{x}_0 repräsentierte Anfangsbedingung berücksichtigt. Wird das Zeitintervall, in dem das Netz $\Psi_0^{\mathcal{W}_0, \mathcal{B}_0}$ die Lösung der partiellen Differentialgleichung approximiert, auf $T \geq \Delta t$ festgelegt, kann die Lösung an den Stützpunkten zum Zeitpunkt Δt mit

$$h(x_i, \Delta t) \approx \Psi_0^{\mathcal{W}_0, \mathcal{B}_0}(\mathbf{X}_{i,\Delta t}) \quad \text{mit} \quad \mathbf{X}_{i,\Delta t} = [x_i, \Delta t]^T \quad (3.57)$$

angenähert werden. Diese Lösungen können als neuer Zustand des Systems zum Zeitpunkt Δt betrachtet werden:

$$\mathbf{x}_1 = \begin{bmatrix} h(x_1, \Delta t) \\ \vdots \\ h(x_i, \Delta t) \\ \vdots \\ h(x_{n_\omega}, \Delta t) \end{bmatrix} \approx \begin{bmatrix} \Psi_0^{\mathcal{W}_0, \mathcal{B}_0}(x_1, \Delta t) \\ \vdots \\ \Psi_0^{\mathcal{W}_0, \mathcal{B}_0}(x_i, \Delta t) \\ \vdots \\ \Psi_0^{\mathcal{W}_0, \mathcal{B}_0}(x_{n_\omega}, \Delta t) \end{bmatrix} \quad . \quad (3.58)$$

Mit diesem Zustand könnte nun ein neues Netz $\Psi_1^{\mathcal{W}_1, \mathcal{B}_1}$ trainiert werden, welches wie im vorherigen Schritt \mathbf{x}_1 über eine Interpolation als Anfangsbedingung berücksichtigt. Das Training benötigt allerdings einen großen, nicht deterministischen Rechenaufwand, wodurch der Ansatz wiederum nicht für eine Echtzeitanwendung geeignet ist.

3 Modellierung

Für eine echtzeitfähige Umsetzung muss ein Netz antrainiert werden, das für beliebige Anfangsbedingungen bzw. Zustände eine Approximation der entsprechenden Lösung der partiellen Differentialgleichung liefert. Das heißt, die Anfangsbedingung muss über einen zusätzlichen Eingang in das Netz explizit berücksichtigt werden:

$$h_{x_l}(x, t) \approx \Psi^{\mathcal{W}, \mathcal{B}}(\mathbf{X}_{x_l}) \quad \text{mit} \quad \mathbf{X}_{x_l} = [x, t, \mathbf{x}_l^T]^T. \quad (3.59)$$

Der Index x_l weist in dieser Notation auf eine Anfangsbedingung hin, die durch einen Zustand \mathbf{x}_l definiert ist. Sollen n_0 Anfangsbedingungen im Training berücksichtigt werden, ergeben sich die angepassten Trainingsmengen

$$\begin{aligned} \mathcal{X}_{T,h} = \{ & \mathbf{X}_{T,h}^{1,1}, \dots, \mathbf{X}_{T,h}^{i_h,1}, \dots, \mathbf{X}_{T,h}^{n_{Th},1}, \dots, \\ & \mathbf{X}_{T,h}^{1,i_0}, \dots, \mathbf{X}_{T,h}^{i_h,i_0}, \dots, \mathbf{X}_{T,h}^{n_{Th},i_0}, \dots, \\ & \mathbf{X}_{T,h}^{1,n_0}, \dots, \mathbf{X}_{T,h}^{i_h,n_0}, \dots, \mathbf{X}_{T,h}^{n_{Th},n_0} \} \quad \text{mit} \quad \mathbf{X}_{T,h}^{i_h,i_0} = [x_{i_h}, t_{i_h}, \mathbf{x}_{i_0}^T]^T \end{aligned} \quad (3.60)$$

$$\begin{aligned} \mathcal{Y}_{T,h} = \{ & \mathbf{Y}_{T,h}^{1,1}, \dots, \mathbf{Y}_{T,h}^{i_h,1}, \dots, \mathbf{Y}_{T,h}^{n_{Th},1}, \dots, \\ & \mathbf{Y}_{T,h}^{1,i_0}, \dots, \mathbf{Y}_{T,h}^{i_h,i_0}, \dots, \mathbf{Y}_{T,h}^{n_{Th},i_0}, \dots, \\ & \mathbf{Y}_{T,h}^{1,n_0}, \dots, \mathbf{Y}_{T,h}^{i_h,n_0}, \dots, \mathbf{Y}_{T,h}^{n_{Th},n_0} \} \quad \text{mit} \quad \mathbf{Y}_{T,h}^{i_h,i_0} = h_{x_{i_0}}(x_{i_h}, t_{i_h}) \end{aligned} \quad (3.61)$$

für die Punkte mit bekannten Lösungen der partiellen Differentialgleichungen,

$$\begin{aligned} \mathcal{X}_{T,\theta} = \{ & \mathbf{X}_{T,\theta}^{1,1}, \dots, \mathbf{X}_{T,\theta}^{i_h,1}, \dots, \mathbf{X}_{T,\theta}^{n_{T\theta},1}, \dots, \\ & \mathbf{X}_{T,\theta}^{1,i_0}, \dots, \mathbf{X}_{T,\theta}^{i_h,i_0}, \dots, \mathbf{X}_{T,\theta}^{n_{T\theta},i_0}, \dots, \\ & \mathbf{X}_{T,\theta}^{1,n_0}, \dots, \mathbf{X}_{T,\theta}^{i_h,n_0}, \dots, \mathbf{X}_{T,\theta}^{n_{T\theta},n_0} \} \quad \text{mit} \quad \mathbf{X}_{T,\theta}^{i_h,i_0} = [x_{i_h}, t_{i_h}, \mathbf{x}_{i_0}^T]^T \end{aligned} \quad (3.62)$$

für die Kollokationspunkte, an denen die partielle Differentialgleichung erfüllt sein soll, und

$$\begin{aligned} \mathcal{X}_{T,h_0} = \{ & \mathbf{X}_{T,h_0}^{1,1}, \dots, \mathbf{X}_{T,h_0}^{i_h,1}, \dots, \mathbf{X}_{T,h_0}^{n_{Th_0},1}, \dots, \\ & \mathbf{X}_{T,h_0}^{1,i_0}, \dots, \mathbf{X}_{T,h_0}^{i_h,i_0}, \dots, \mathbf{X}_{T,h_0}^{n_{Th_0},i_0}, \dots, \\ & \mathbf{X}_{T,h_0}^{1,n_0}, \dots, \mathbf{X}_{T,h_0}^{i_h,n_0}, \dots, \mathbf{X}_{T,h_0}^{n_{Th_0},n_0} \} \quad \text{mit} \quad \mathbf{X}_{T,h_0}^{i_h,i_0} = [x_{i_h}, 0, \mathbf{x}_{i_0}^T]^T \end{aligned} \quad (3.63)$$

$$\begin{aligned} \mathcal{Y}_{T,h_0} = \{ & \mathbf{Y}_{T,h_0}^{1,1}, \dots, \mathbf{Y}_{T,h_0}^{i_h,1}, \dots, \mathbf{Y}_{T,h_0}^{n_{Th_0},1}, \dots, \\ & \mathbf{Y}_{T,h_0}^{1,i_0}, \dots, \mathbf{Y}_{T,h_0}^{i_h,i_0}, \dots, \mathbf{Y}_{T,h_0}^{n_{Th_0},i_0}, \dots, \\ & \mathbf{Y}_{T,h_0}^{1,n_0}, \dots, \mathbf{Y}_{T,h_0}^{i_h,n_0}, \dots, \mathbf{Y}_{T,h_0}^{n_{Th_0},n_0} \} \quad \text{mit} \quad \mathbf{Y}_{T,h_0}^{i_h,i_0} = \xi(x_{i_h}, \mathbf{x}_{i_0}, \omega) \end{aligned} \quad (3.64)$$

für die Berücksichtigung der Punkte, die zur Anfangsbedingung gehören. Diese Trainingsdaten können dann über die entsprechende Trainingsfunktion in Gl. (3.56) berücksichtigt werden, um ein Netz $\Psi^{\mathcal{W}, \mathcal{B}}$ zu trainieren, das die Lösung der partiellen Differentialgleichung für eine vorgegebene Anfangsbedingung approximiert. Mit diesem Ansatz ist es nun möglich, die Lösung der partiellen Differentialgleichung über einen beliebigen Zeithorizont zu approximieren. Dafür wird ausgehend vom aktuellen Zustand zum Zeitpunkt t_l das Netz genutzt, um den Zustand zum Zeitpunkt $t_l + \Delta t$ zu bestimmen. Dieser kann dann im nächsten Zeitschritt wieder als neuer Eingang in das Netz genutzt werden, um eine Approximation für das nächste Zeitintervall zu erhalten. Es ergibt sich ein iteratives Vorgehen.

Für die Berücksichtigung der Aktuation muss zunächst definiert werden, in welcher Weise das System von außen beeinflusst werden kann bzw. beeinflusst werden soll. Eine naheliegende

Lösung ist eine Aktuierung über zeitlich veränderliche Randbedingungen oder das Einbringen eines variablen Quellterms. Für den Fall einer Aktuierung über eine Randbedingung ist prinzipiell ein ähnliches Vorgehen wie für die Anfangsbedingung möglich. Allerdings haben Versuche zu diesem Ansatz gezeigt, dass die Berücksichtigung der Aktuierung über einen zusätzlichen Eingang selbst bei deutlich vergrößerter Netzstruktur und Trainingsmenge zu deutlich schlechteren Approximationsergebnissen führt.

Daher wird an dieser Stelle ein alternativer Weg gewählt. Die Aktuierung über einen Zeitschritt Δt ist als konstant angenommen und wird im Folgenden durch die skalare Größe μ_l beschrieben. Für reale Anwendungen kann zudem von einer beschränkten Aktuierung

$$\mu_{lb} \leq \mu_l \leq \mu_{ub} \quad (3.65)$$

ausgegangen werden. Die Aktuierung kann nun in eine endliche Anzahl $n_{\mathcal{M}}$ an Schritten diskretisiert werden. Für eine äquidistante Diskretisierung ergibt sich

$$\begin{aligned} \mathcal{M} &= \{\mu_{lb}, \mu_{lb} + \frac{\mu_{ub} - \mu_{lb}}{n_{\mathcal{M}} - 1}, \mu_{lb} + 2\frac{\mu_{ub} - \mu_{lb}}{n_{\mathcal{M}} - 1}, \dots, \mu_{ub}\} \\ &= \{\mu_1, \dots, \mu_i, \dots, \mu_{n_{\mathcal{M}}}\} \end{aligned} \quad (3.66)$$

als die Menge aller Aktuierungsmöglichkeiten. Für jeden einzelnen Aktuierungsschritt aus der Menge \mathcal{M} kann ein Netz Ψ_{μ_i} nach dem beschriebenen Vorgehen trainiert werden, da für die Dauer Δt des einzelnen Zeitschrittes die Aktuierung bzw. die Randbedingungen konstant sind. Es ergibt sich die Menge an Netzen

$$\mathcal{F} = \{\Psi_{\mu_1}(\mathbf{X}), \dots, \Psi_{\mu_i}(\mathbf{X}), \dots, \Psi_{\mu_{n_{\mathcal{M}}}}(\mathbf{X})\} \quad \text{mit} \quad \mathbf{X} = [\mathbf{x}, t, \mathbf{x}^T]^T \quad (3.67)$$

Um auch eine Aktuierung μ_l zu berücksichtigen, die nicht in der Menge \mathcal{M} enthalten ist, kann ein Interpolationsoperator

$$\xi_{\Psi}(\Psi_{\overline{\mu_l}}(\mathbf{X}), \Psi_{\underline{\mu_l}}(\mathbf{X})) \quad (3.68)$$

definiert werden.⁴ Dabei bezeichnen $\overline{\mu_l}$ und $\underline{\mu_l}$ die nächstgrößere bzw. nächstkleinere Aktuierung, die in der Menge \mathcal{M} enthalten ist und für die folglich ein antrainiertes Netz in \mathcal{F} existiert. Damit kann in Abhängigkeit der Zeitschrittweite Δt und der Zustandsdefinition ein diskretes Zustandsraummodell

$$\mathbf{x}_{l+1} = \mathbf{f}(\mathbf{x}_l, \mu_l) = \begin{bmatrix} \xi_{\Psi}(\Psi_{\overline{\mu_l}}(\mathbf{X}_1), \Psi_{\underline{\mu_l}}(\mathbf{X}_1)) \\ \vdots \\ \xi_{\Psi}(\Psi_{\overline{\mu_l}}(\mathbf{X}_i), \Psi_{\underline{\mu_l}}(\mathbf{X}_i)) \\ \vdots \\ \xi_{\Psi}(\Psi_{\overline{\mu_l}}(\mathbf{X}_{n_{\omega}}), \Psi_{\underline{\mu_l}}(\mathbf{X}_{n_{\omega}})) \end{bmatrix} \quad \text{mit} \quad \mathbf{X}_i = [\mathbf{x}_i, \Delta t, \mathbf{x}_i^T]^T \quad (3.69)$$

$$\gamma_l = \mathbf{z}(\mathbf{x}_l, \mu_l) \quad (3.70)$$

aufgebaut werden. Die Messgleichung ist an dieser Stelle nicht genauer definiert, da sie von der konkreten Anwendung abhängt.

In den folgenden Abschnitten wird für die Burgers-Gleichung und die Euler-Gleichungen der Aufbau eines Zustandsraummodells auf Basis des hier beschriebenen PINN-Ansatzes diskutiert. Diese Zustandsraummodelle werden im Anschluss in Anwendungsbeispielen im Kontext einer Zustandsschätzung bzw. einer modellprädiktiven Regelung verwendet.

⁴Für eine digitale Umsetzung kann mitunter auf die Interpolation verzichtet werden, solange die Menge der trainierten Netze passend zur Diskretisierung auf der verwendeten Hardware gewählt wird.

3.4.1 PINN-Approximation der Burgers-Gleichung

Um aufzuzeigen, wie sich die Berücksichtigung eines Zustandes, d.h. einer variablen Anfangsbedingung, auf die Trainingsergebnisse auswirkt, wird an dieser Stelle nochmal das Beispiel aus Abschnitt 3.3 aufgegriffen. Im Training wurden anstelle der sinusförmigen Anfangsbedingung aus Gl. (3.45) 100 zufällige Verläufe der Anfangsbedingung berücksichtigt, welche durch einen entsprechenden Zustand aus dem Bereich $-1 \leq \mathbf{x}_i \leq 1$ repräsentiert werden können. Für jeden dieser 100 Fälle wurden Trainingsdaten an den gleichen 6203 Punkten für Rand- und Anfangsbedingung sowie den Kollokationspunkten der partiellen Differentialgleichung trainiert. Die Punkte entsprechen dabei denen aus Abschnitt 3.3. Damit wächst die Menge der Eingangstrainingsdaten auf knapp über 600 000 an. Dauerte das Training wie in Abschnitt 3.3 beschrieben, ohne Zustand als Netzeingang, 1-2 Minuten, so erhöht sich die Trainingszeit auf gleicher Hardware und bei gleicher Netzgröße auf etwa 6-7 Stunden.⁵ Im Hinblick auf die Anwendbarkeit in einem Aufbau mit Echtzeitanforderungen ist dieser Zuwachs an Rechenaufwand als unkritisch zu betrachten, da das Training ausschließlich einmal vor der eigentlichen Anwendung erfolgen muss.

Ähnlich zu Abb. 3.4 aus Abschnitt 3.3, zeigt Abb. 3.6 oben erneut einen Intensitätsplot für die Netzauswertung auf einem Raum-Zeit-Gitter mit 500 äquidistanten räumlichen Punkten und 200 äquidistanten zeitlichen Punkten. Im direkten Vergleich mit dem in Abb. 3.4 gezeigten

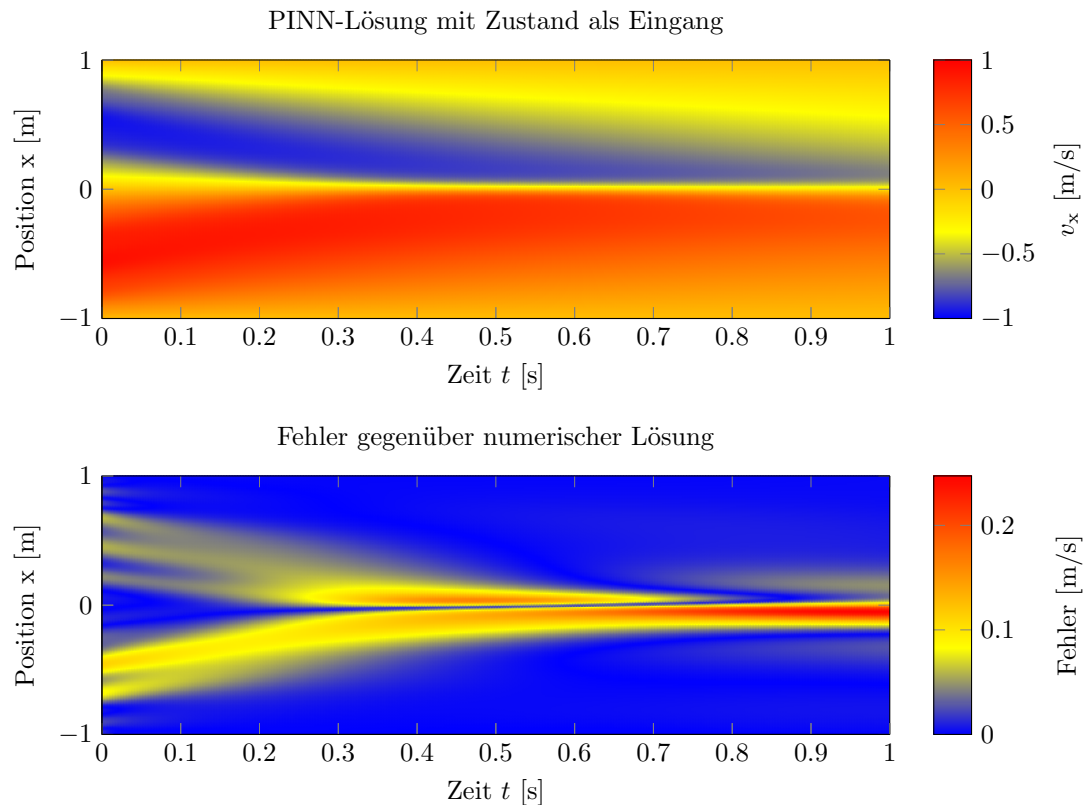


Abbildung 3.6: PINN-Lösung aus einem Training, in dem 100 zufällige Zustände als Repräsentation der Anfangsbedingungen berücksichtigt wurden. Die tatsächliche, hier verwendete Anfangsbedingung ist nicht im Training enthalten.

Oben: Intensitätsplot in der Raum-Zeit-Ebene

Unten: Fehler gegenüber einer numerischen Lösung mit einer FVM

⁵Es wurde eine GeForce Titan V GPU verwendet.

3.4 Zustandsraummodellierung auf Basis von Physics-Informed-Neural-Networks

Ergebnis ist ein Unterschied kaum zu erkennen. Die Betrachtung der Abweichung gegenüber der numerischen Lösung, zu sehen in Abb. 3.6 unten, zeigt jedoch, dass für den Fall mit einem Zustand als Netzeingang eine um eine Größenordnung gestiegene Abweichung auftritt. Dieses Ergebnis ist auch in Abb. 3.7 zu sehen, in der die Lösungen zu verschiedenen Zeitpunkten über die räumliche Koordinate x dargestellt sind.

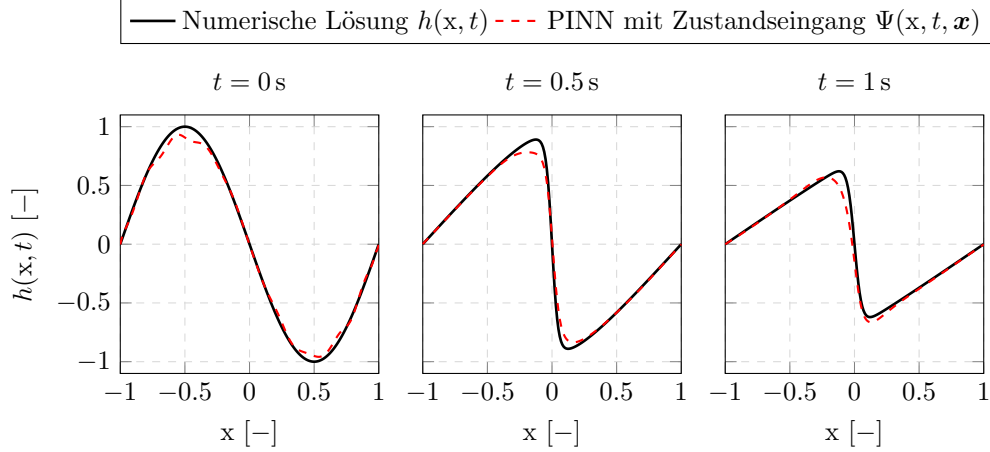


Abbildung 3.7: Vergleich zwischen PINN-Lösung aus einem Training mit 100 zufälligen Zuständen als Repräsentation der Anfangsbedingung und einer numerischen Lösung mit einer FVM zu den Zeitpunkten $t = 0$ s, $t = 0.5$ s und $t = 1$ s. Die tatsächliche Anfangsbedingung ist nicht in der Trainingsmenge enthalten.

Trotz der merklichen Abweichung kann das Ergebnis weiterhin als zufriedenstellende Approximation betrachtet werden. Die wichtigsten Eigenschaften des dynamischen Verhaltens werden nach wie vor gut angenähert. Zudem sollte berücksichtigt werden, dass die gezeigte PINN-Approximation in einer Regelung bzw. Zustandsschätzung Verwendung finden soll. In diesen Fällen kann davon ausgegangen werden, dass Modellfehler dieser Art kompensiert werden können.

3.4.2 PINN-Approximation der Euler-Gleichungen

Die Anpassung eines geeigneten Zustandsraummodells für die Euler-Gleichungen auf Basis der PINN-Modellierung ist deutlich komplexer als für die Burgers-Gleichung. Der Grund dafür ist, dass nicht nur eine einzelne Differentialgleichung das System beschreibt, sondern, wie in Abschnitt 3.1.2 ausgeführt, die Strömung durch ein Gleichungssystem aus drei partiellen Differentialgleichungen und einer algebraischen Gleichung besteht.

Der Zustand des durch die Euler-Gleichungen beschriebenen Systems ist zu jedem Zeitpunkt und an jeder Position definiert über die Dichte $\rho(x, t)$, die Geschwindigkeit $v_x(x, t)$ und die innere Energie $E(x, t)$.⁶ Die Lösung des partiellen Differentialgleichungssystems

$$\mathbf{h}(x, t) = [\rho(x, t), v_x(x, t), E(x, t)]^T \quad (3.71)$$

für eine Anfangsbedingung

$$\mathbf{h}(x, 0) = [\rho(x, 0), v_x(x, 0), E(x, 0)]^T \quad (3.72)$$

⁶Eine Definition über andere Kombinationen von Größen, beispielsweise Druck, Dichte und Geschwindigkeit, ist ebenfalls möglich und kann je nach Anwendung Vorteile in der Umsetzung bieten.

3 Modellierung

ist folglich vektoriell und besteht aus drei Komponenten. Wie in Abschnitt 3.4 beschrieben, wird die Anfangsbedingung für das Training durch einen diskreten Zustand repräsentiert. Dafür wird wie oben ausgeführt die Anfangsbedingung an n_ω Stellen ausgewertet und zum Zustand

$$\mathbf{x}_0 = \begin{bmatrix} \mathbf{h}(\mathbf{x}_1, 0) \\ \vdots \\ \mathbf{h}(\mathbf{x}_{n_\omega}, 0) \end{bmatrix} = \begin{bmatrix} \rho(\mathbf{x}_1, 0) \\ v_x(\mathbf{x}_1, 0) \\ E(\mathbf{x}_1, 0) \\ \vdots \\ \rho(\mathbf{x}_{n_\omega}, 0) \\ v_x(\mathbf{x}_{n_\omega}, 0) \\ E(\mathbf{x}_{n_\omega}, 0) \end{bmatrix} \quad (3.73)$$

zusammengeführt. Für die Approximation der Lösung $\mathbf{h}(\mathbf{x}, t)$ mittels eines PINNs bieten sich nun mehrere Möglichkeiten. In Raissi, Perdikaris und Karniadakis (2019) und Raissi, Wang, Triantafyllou und Karniadakis (2019) wird für die Approximation einer Lösung der Navier-Stokes-Gleichungen ein einziges neuronales Netz verwendet, welches das Geschwindigkeitsfeld in Abhängigkeit von Ort und Zeit beschreiben soll. Allerdings wird in den genannten Arbeiten auch keine variable Anfangsbedingung berücksichtigt. Zudem wird das Fluid als inkompressibel angenommen, was eine Approximation deutlich erleichtert. Für den hier relevanten Fall mit variabler Anfangsbedingung repräsentiert durch einen Zustand \mathbf{x} müsste das Netz auf eine Abbildung

$$\Psi(\mathbf{x}, t, \mathbf{x}) : \mathbf{x}, t, \mathbf{x} \rightarrow \rho, v_x, E \quad (3.74)$$

trainiert werden. Abweichend davon wird in diesem Abschnitt ein Weg über drei voneinander separate neuronale Netze gewählt, welche über die partiellen Differentialgleichungen im Training gekoppelt sind. Mit drei Netzen werden folglich drei Abbildungen

$$\Psi_\rho(\mathbf{x}, t, \mathbf{x}) : \mathbf{x}, t, \mathbf{x} \rightarrow \rho, \quad (3.75)$$

$$\Psi_{v_x}(\mathbf{x}, t, \mathbf{x}) : \mathbf{x}, t, \mathbf{x} \rightarrow v_x \quad (3.76)$$

$$\Psi_E(\mathbf{x}, t, \mathbf{x}) : \mathbf{x}, t, \mathbf{x} \rightarrow E \quad (3.77)$$

erzeugt. Der Ansatz über drei separate Netze bietet erhebliche Vorteile in der weiter unten beschriebenen Vorkonditionierung, da so Probleme in der Normierung für das Training der Netze umgangen werden können. Zusätzlich bietet dieser Ansatz in der Verwendung des Modells die Möglichkeit, für die Bestimmung einzelner Werte an unterschiedlichen Stellen nur das relevante Netz auszuwerten.

Wie in Abschnitt 3.4 beschrieben und im Beispiel der Burgers-Gleichung gezeigt, werden auch für die Euler-Gleichungen die Trainingsdaten in drei Gruppen unterteilt:

- Die Anfangsbedingung wird in der Trainingsmenge $\mathcal{X}_{T, h_0}, \mathcal{Y}_{T, h_0}$ mit der Trainingsfunktion gemäß Gl. (3.55) berücksichtigt. Die Anfangsbedingung wird durch den Zustand \mathbf{x} repräsentiert.
- Die Punkte, an denen die Lösung der partiellen Differentialgleichung bekannt ist, z.B. an den Rändern, werden in der Trainingsmenge $\mathcal{X}_{T, h}, \mathcal{Y}_{T, h}$ berücksichtigt.
- Die Kollokationspunkte für das Training der partiellen Differentialgleichungen gemäß Gl. (3.43) werden in der Trainingsmenge $\mathcal{X}_{T, \theta}$ berücksichtigt.

3.4 Zustandsraummodellierung auf Basis von Physics-Informed-Neural-Networks

Der Unterschied gegenüber der Burgers-Gleichung ist die Dimension der Lösung. Für die Elemente der Trainingsmengen $\mathcal{Y}_{T,h}$ gilt

$$\mathbf{Y}_{T,h}^{i_h,i_0} = \mathbf{h}_{x_{i_0}}(x_{i_h}, t_{i_h}) = \left[\rho_{x_{i_0}}(x_{i_h}, t_{i_h}), v_{x,x_{i_0}}(x_{i_h}, t_{i_h}), E_{x_{i_0}}(x_{i_h}, t_{i_h}) \right]^T \quad (3.78)$$

Entsprechend muss die Interpolation Gl. (3.68) auch vektoriell ausgeführt werden. Die Elemente der Trainingsmenge \mathcal{Y}_{T,h_0} ergeben sich dann zu

$$\mathbf{Y}_{T,h_0}^{i_h,i_0} = \boldsymbol{\xi}(x_{i_h}, x_{i_0}, \boldsymbol{\omega}) \quad (3.79)$$

Die jeweils drei Komponenten der Elemente in den Trainingsmengen \mathcal{Y}_{T,h_0} und $\mathcal{Y}_{T,h}$ werden separat für das Training der drei Netze $\Psi_\rho, \Psi_{v_x}, \Psi_E$ genutzt. An den Kollokationspunkten der Trainingsmenge $\mathcal{X}_{T,\theta}$ wird dann für die Ableitungen der Netze gefordert

$$\frac{\partial}{\partial t} \begin{bmatrix} \Psi_\rho \\ \Psi_\rho \Psi_{v_x} \\ \Psi_\rho \Psi_E \end{bmatrix} + \frac{\partial}{\partial x} \begin{bmatrix} \Psi_\rho \Psi_{v_x} \\ \Psi_\rho \Psi_{v_x}^2 + p \\ \Psi_{v_x}(\Psi_\rho \Psi_E + p) \end{bmatrix} = \begin{bmatrix} \theta_1 \\ \theta_2 \\ \theta_3 \end{bmatrix} \stackrel{!}{=} \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \quad (3.80)$$

Der Druck p wird wie in der numerischen Lösung über die Annahme eines idealen Gases gemäß Gl. (3.15) ermittelt. Mit den drei Netzen folgt für den Druck

$$p \approx \left(\Psi_E + \frac{1}{2} \Psi_{v_x}^2 \right) (\gamma - 1) \Psi_\rho \quad (3.81)$$

Dieser Zusammenhang kann anschließend in der zweiten und dritten Komponente von Gl. (3.80) genutzt werden. Die Ausdrücke θ_1, θ_2 und θ_3 können schließlich in der Optimierung der Gewichte berücksichtigt werden analog zu Gl. (3.43).

Mit den entsprechenden Trainingsdaten ist ein Trainingsprozess genau so wie für das Beispiel der Burgers-Gleichungen möglich. Allerdings haben Versuche mit verschiedenen Netztopologien gezeigt, dass der Trainingsprozess häufig in verschiedenen lokalen Minima endet, was die Qualität der trainierten Abbildung stark variieren lässt. In etwa der Hälfte der Trainingsprozesse ergibt sich für die Euler-Gleichungen ein antrainiertes Netz, welches in der Qualität so schlecht ist, dass die groben zeitlichen Verläufe der Größen nicht richtig abgebildet werden können. Aus diesem Grund ist für den Fall der Euler-Gleichungen eine Anpassung des Trainingsprozesses notwendig, um ein zuverlässig gutes Ergebnis sicher zu stellen. Im Gegensatz zu dem ursprünglich präsentierten PINN-Ansatz in Raissi, 2018 ist die vorgestellte Methode für die Zustandsraummodellierung keine Alternative zur numerischen Lösung des partiellen Differentialgleichungssystems. Vielmehr ist davon auszugehen, dass der Arbeitsablauf für den Aufbau einer Strömungsregelung aus folgenden Schritten besteht:

- Es wird eine numerische Simulation des Systems auf Basis partieller Differentialgleichungen aufgebaut, an der theoretische Untersuchungen durchgeführt werden können.
- Die Simulation wird mit Daten von experimentellen Untersuchungen abgeglichen, um zu überprüfen, ob die numerische Lösung die Realität ausreichend gut abbildet.
- Es wird ein Zustandsraummodell auf PINN-Basis trainiert und simulativ für die entsprechende Regelungs- oder Zustandsschätzungsanwendung getestet.
- Die PINN-basierte Regelung oder Zustandsschätzung wird auf das reale System angewendet.

3 Modellierung

In einem solchen Arbeitsablauf liegen während des Aufbaus des PINN-Netzes bereits numerische Daten vor, die das reale Experiment zufriedenstellend gut abbilden. Wie bereits beschrieben, ist der Anspruch des PINN-Zustandsraummodells, eine recheneffiziente Approximation der Lösung der partiellen Differentialgleichungen bereitzustellen, die in Echtzeit genutzt werden kann. So muss die volle numerische Lösung nicht berechnet werden. In diesem Sinne ist das PINN-basierte Zustandsraummodell als eine Art Modellreduktion zu sehen. Daher ist es durchaus sinnvoll, gewonnene Informationen aus der numerischen Simulation im Training der neuronalen Netze zu berücksichtigen.

Um zu vermeiden, dass der Trainingsprozess in ein besonders ungünstiges lokales Minimum konvergiert, wird ein zweistufiger Trainingsprozess eingeführt, der im ersten Schritt die vorhandenen Daten aus einer numerischen Simulation nutzt. Eine schematische Darstellung des zweistufigen Trainingsprozesses ist in Abb. 3.8 dargestellt. Dieser erste Schritt wird im Folgenden als Vorkonditionierung bezeichnet. Es wird angenommen, dass die numerische Lösung in guter Näherung das partielle Differentialgleichungssystem erfüllt. Daher wird in der Vorkonditionierung die Information aus der Differentialgleichung im Training der Netze ausgelassen. Es wird also ausschließlich auf den Trainingsmengen \mathcal{X}_{T,h_0} , \mathcal{Y}_{T,h_0} und $\mathcal{X}_{T,h}$, $\mathcal{Y}_{T,h}$ trainiert. Allerdings wird die numerische Lösung an n_N Punkten der Trainingsmenge $\mathcal{X}_{T,h}$, $\mathcal{Y}_{T,h}$ hinzugefügt. Zur besseren Unterscheidung wird diese erweiterte Trainingsmenge hier mit $\tilde{\mathcal{X}}_{T,h}$, $\tilde{\mathcal{Y}}_{T,h}$ bezeichnet. Dies hat zur Folge, dass in der Vorkonditionierung die drei Netze unabhängig voneinander trainiert werden, da die Kopplung nur über die Berücksichtigung der partiellen Differentialgleichungen entsteht. Da jedoch die Daten der numerischen Lösung berücksichtigt werden, kann davon ausgegangen werden, dass die optimierten Netze nach dem Training das Differentialgleichungssystem bereits in Näherung erfüllen. Um jedoch das gewünschte robuste Verhalten des PINN-Modells sicherzustellen, wird in der zweiten Phase die Information der Differentialgleichungen, wie im ursprünglichen Ansatz des PINN-Zustandsraummodells vorgesehen, mit berücksichtigt. Dafür werden die n_N Punkte der numerischen Lösung nicht mehr in der Trainingsmenge $\mathcal{X}_{T,h}$, $\mathcal{Y}_{T,h}$ berücksichtigt. Ausschließlich die Punkte der Randbedingungen werden weiter berücksichtigt. Stattdessen werden die Kollokationspunkte aus $\mathcal{X}_{T,\theta}$ in der zweiten Phase für das Training des Zusammenhangs aus den partiellen Differentialgleichungen einbezogen.

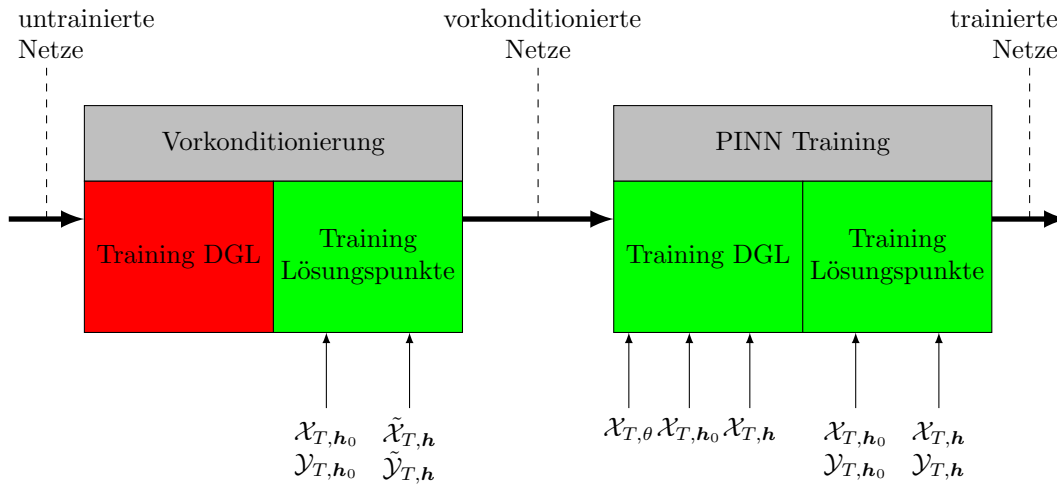


Abbildung 3.8: Ablauf des zweistufigen Trainingsprozess eines PINN-Modells zur Approximation der Euler-Gleichungen.

Um den Trainingsverlauf zu veranschaulichen, sind in Abb. 3.9 die mittleren quadratischen

3.4 Zustandsraummodellierung auf Basis von Physics-Informed-Neural-Networks

Fehler gegenüber der numerischen Lösung der drei Netze exemplarisch für beide Trainingsphasen dargestellt. Es werden drei „Feed Forward“-Netze mit acht verborgenen Schichten mit

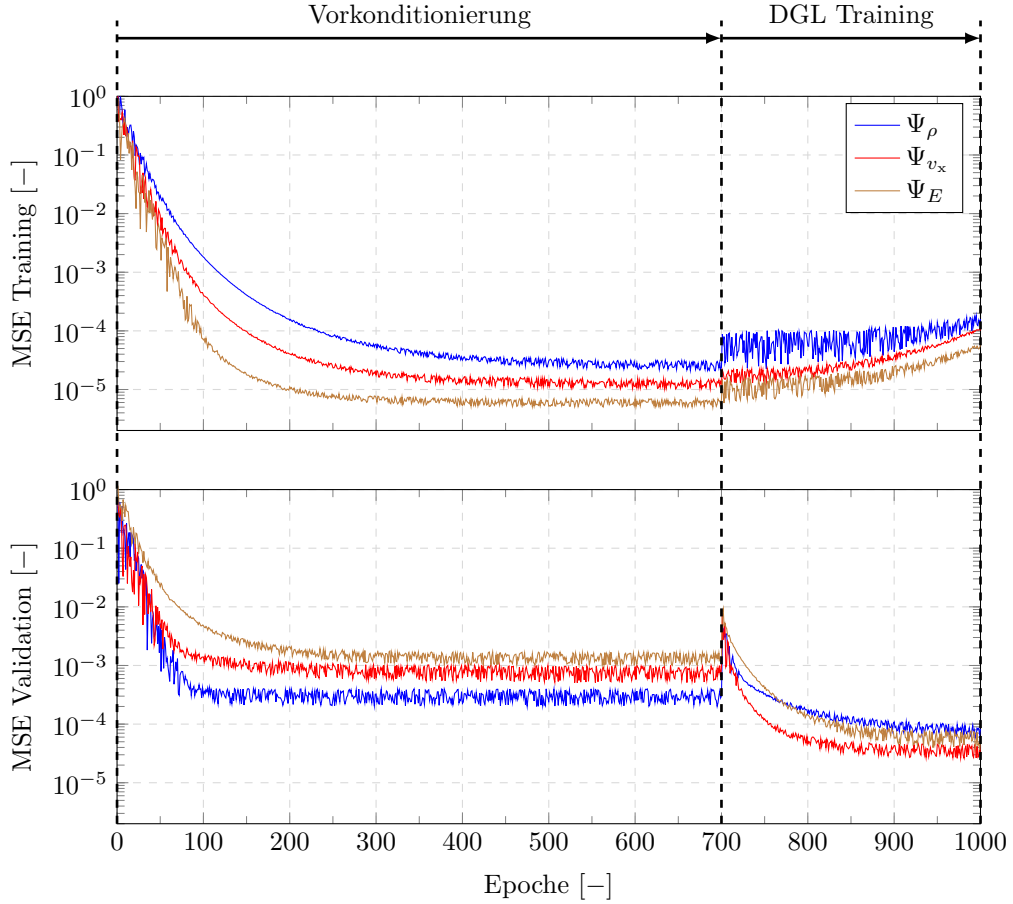


Abbildung 3.9: Beispielhafte Progression der Kostenfunktionale (*Mean Squared Error*, MSE) im zweistufigen Training eines PINN-Modells. Es werden drei Netze für Dichte, Geschwindigkeit und Energie antrainiert. Die ersten 700 Epochen zeigen die Vorkonditionierung mit numerischen Daten. Die folgenden 300 Epochen zeigen das Training mit Anfangs- und Randbedingungen und den Informationen aus dem partiellen Differentialgleichungssystem.

jeweils 50 Neuronen verwendet. Zusätzlich hat jedes Netz eine Eingangsschicht mit Neuronen entsprechend der Zustandsdimension und eine Ausgangsschicht mit einem einzelnen Neuron. In diesem Trainingsbeispiel wurden die Trainings- und Validationsdaten im Verhältnis 80 zu 20 geteilt. Für 700 Epochen wurden die drei Netze unabhängig voneinander mit den Daten aus der numerischen Lösung vorkonditioniert. Im Anschluss wurde für 300 Epochen mit den Anfangs- und Randbedingungen sowie den Informationen aus den Differentialgleichungen gemäß den Gl. (3.80) mit der entsprechenden Kostenfunktion Gl. (3.56) trainiert.

Hier zeigt sich, dass sich die Prädiktionsgüte der Netze für die Validationsdaten bereits nach etwa 200 Epochen der Vorkonditionierung nicht mehr deutlich verbessert. Insgesamt liegt am Ende der Vorkonditionierung das Niveau des Fehlers in Bezug auf die Validationsdaten ein bis zwei Größenordnungen über dem Niveau der Trainingsdaten. Dies deutet auf eine Überanpassung der Netze in der Vorkonditionierung hin. In der zweiten Phase, in der das partielle Differentialgleichungssystem berücksichtigt wird, verschlechtert sich die Performance zunächst sowohl für die Validations- als auch die eigentlichen Trainingsdaten. Im Verlauf der 300 Epochen gleichen sich die Niveaus dann allerdings an, bis sich schließlich ein ähnliches

3 Modellierung

Niveau des Fehlers für Validations- und Trainingsdaten ergibt. Der Verlauf der Fehler zeigt hier eindeutig, wie das Training mit den Informationen des Differentialgleichungssystems die Ergebnisse für die Validationsdaten verbessert. Das ähnliche Fehlerniveau für Trainingsdaten und Validationsdaten deutet darauf hin, dass eine Überanpassung des Modells vermieden wird.

Um die Qualität der Approximation eines PINN-Modells zu veranschaulichen, wird im Folgenden, wie bei der Burgers-Gleichung, ein kurzes Beispiel diskutiert. Der betrachtete Simulationsbereich $x \in [0, 1]$ m kann als ein ein Meter langes Rohr aufgefasst werden. Beide Enden des Rohres werden als geschlossen mit reflektierenden Rändern angenommen. Für das Beispiel wird eine Beschreibung der Anfangsbedingung bzw. des Zustands über die Werte für Dichte, Geschwindigkeit und spezifischen Energie an fünf äquidistanten Stellen angenommen:

$$\mathbf{x} = \begin{bmatrix} \rho(x_1, 0) \\ v_x(x_1, 0) \\ E(x_1, 0) \\ \vdots \\ \rho(x_5, 0) \\ v_x(x_5, 0) \\ E(x_5, 0) \end{bmatrix} \in \mathbb{R}^{15} . \quad (3.82)$$

Da aufgrund der Kompressibilität die auftretenden Strömungsphänomene auf einer deutlich kürzeren Zeitskala stattfinden als bei der Burgers-Gleichung, ist das Training eines neuronalen Netzes auch nur für einen kürzeren Zeitbereich möglich. Im gezeigten Beispiel wurde das neuronale Netz für den Zeitbereich $t \in [0, 1]$ ms antrainiert. Für das Training wurde ein Datensatz erstellt, der 800 zufällige Zustände bzw. Anfangsbedingungen berücksichtigt. Dabei wurden die Zustände so gewählt, dass für das System gilt:

- Die Temperatur liegt im Bereich $\vartheta \in [300, 600]$ K.
- Der Druck liegt im Bereich $p \in [1, 3]$ bar.
- Die Geschwindigkeit liegt im Bereich $v_x \in [1, 3]$ m/s.

Im Training werden Anfangsbedingungen an 500 und Randbedingungen an jeweils 200 äquidistanten Stellen berücksichtigt. In der Vorkonditionierung werden Daten aus der numerischen Lösung auf einem äquidistanten Raum-Zeit-Gitter mit 100×10 Gitterpunkten hinzugefügt. In der zweiten Trainingsphase werden 5000 zufällig verteilte Kollokationspunkte in der Raum-Zeit-Ebene für die Information der partiellen Differentialgleichungen ausgewertet. Für die Netztopologie werden, wie beschrieben, drei Netze mit jeweils acht Schichten aus 50 Neuronen verwendet. Davor ist jeweils eine Schicht mit 17 Neuronen, um die Eingänge x , t und \mathbf{x} zu verarbeiten. Nach der letzten Schicht folgt eine zusätzliche Ausgabeschicht mit einem einzelnen Neuron. Abweichend zur Burgers-Gleichung wird in diesem Beispiel die ReLu-Funktion für die Aktivierung der Neuronen verwendet. In beiden Phasen wird der BFGS-Algorithmus für die Optimierung der Netzparameter genutzt.

Die Abb. 3.10, 3.11 und 3.12 zeigen jeweils die numerische Lösung, die PINN-Lösung und den Fehler zwischen den beiden Lösungen separat für Dichte, Geschwindigkeit und Energie. Dabei wird ein zufälliger Zustandseingang verwendet, der nicht in der Trainingsmenge enthalten ist. Die Ergebnisse wurden an 200 zufälligen Beispielen für den Zustandseingang validiert. Beide Lösungen wurden auf einem Raum-Zeit-Gitter von 100×50 äquidistanten Punkten ausgewertet. Die PINN-Lösung wurde direkt an diesen Stellen ausgewertet, die numerische

Lösung wurde von einer feineren Lösung mit variierenden Zeitabständen auf das entsprechende Gitter linear interpoliert. Dies ist notwendig, da die numerische Simulation des Systems, wie in Abschnitt 3.1.4 beschrieben, mit einer adaptiven Zeitschrittweite arbeitet.

In allen drei Abbildungen lässt sich deutlich erkennen, dass das PINN-Modell in der Lage ist, die signifikanten Charakteristiken in der Raum-Zeit-Ebene qualitativ gut abzubilden. Das beste Ergebnis ergibt sich in diesem Beispiel für die Dichte. Aber auch der zeitliche Verlauf von Geschwindigkeit und Energie wird im Hinblick auf die komplexe Entwicklung über den betrachteten Zeitbereich zufriedenstellend abgebildet. Der relative Fehler bezogen auf die Differenz zwischen maximalem und minimalem Wert in der jeweiligen numerischen Lösung liegt für alle drei betrachteten Größen im Bereich von etwa 0 – 15 %. Die mittleren Werte für diesen relativen Fehler liegen bei etwa 1.3 % für die Dichte, 2.8 % für die Geschwindigkeit und 3.2 % für die Energie. In diesem Beispiel ist es deutlich schwieriger, die Geschwindigkeit und die Energie mit dem PINN-Modell akkurat zu approximieren als die Dichte. Mit einem Blick auf den zeitlichen Verlauf der Dichte in Abb. 3.10 ist dies nachvollziehbar. Für die betrachtete Anfangsbedingung ist die zeitliche Variation der Dichte über die Länge des Rohres relativ gering. Das zeigt sich vor allem im Bereich zwischen $x = 0.5$ und $x = 1$. Ein besonders stark ausgeprägter Fehler zeigt sich in der Energie am Rand, $x = 0$. Dies ist vermutlich auf die schwierige unstetige Abbildung der Druckreflektion am Rand des Systems zurückzuführen.

Die unterschiedliche Qualität in der Approximation zeigt sich auch in Abb. 3.13. Hier sind die räumlichen Verläufe der Dichte, Geschwindigkeit und Energie für die Zeitpunkte $t = 0$ ms, $t = 0.5$ ms, $t = 1$ ms dargestellt. Für die Dichte zeigt sich ein präziser und relativ glatter Verlauf des PINN-Modells im Vergleich zur numerischen Lösung. Im Gegensatz dazu zeigen die Verläufe von Geschwindigkeit und Energie eine stärkere Varianz und unterscheiden sich insgesamt stärker von der numerischen Lösung als im Fall der Dichte.

Auch wenn die Ergebnisse für die Approximation der Euler-Gleichungen nicht die gleiche Qualität aufweisen wie die Resultate für die Burgers-Gleichung, ist die Performance des PINN-Modells insgesamt als gut und ausreichend zu bewerten. Dies trifft vor allem mit Blick auf die Verwendung in der Zustandsschätzung oder der modellbasierten Regelung zu, bei der die Ansätze so konzipiert sind, dass Modellfehler bis zu einem gewissen Maß korrigiert werden können. Da, wie beschrieben, die wichtigsten dynamischen Charakteristiken des Systems gut abgebildet werden, ist davon auszugehen, dass die PINN-Modellierung gute Ergebnisse in den entsprechenden Anwendungen liefert. Um dies zu verdeutlichen werden, in den nächsten beiden Kapiteln Ergebnisse für eine Zustandsschätzung und eine MPC auf Basis des PINN-Modells gezeigt.

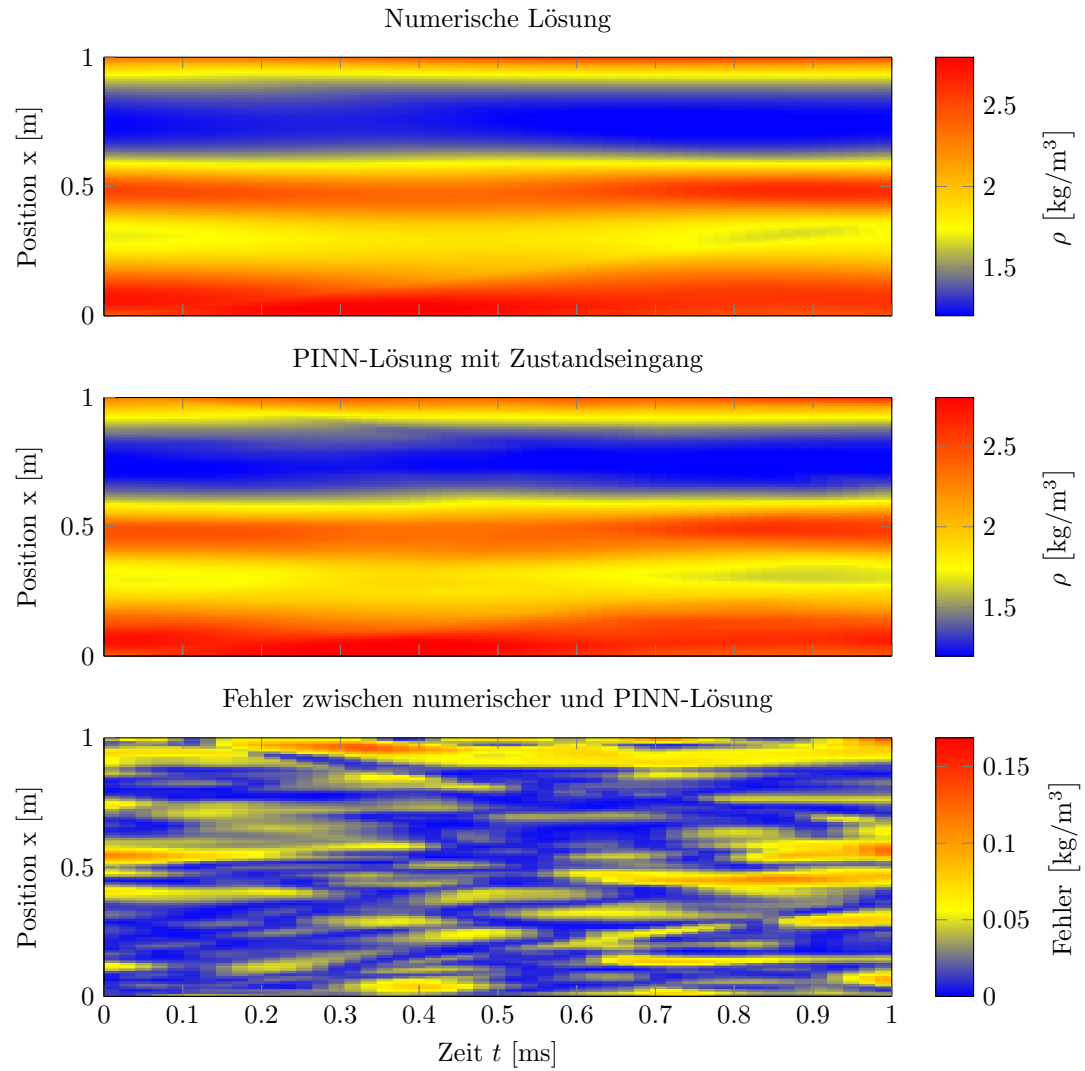


Abbildung 3.10: Vergleich des Verlaufs der Dichte aus einer numerischen Lösung und der PINN-Lösung mit Zustandseingang.

Oben: Numerische Lösung mit einer FVM gemäß der Beschreibung in Abschnitt 3.1.4.

Mitte: PINN-Lösung mit Zustandseingang; die Zustandsrepräsentation entspricht den Dichte-, Geschwindigkeits- und Energiewerten an fünf äquidistanten Stützstellen.

Unten: Absoluter Fehler zwischen den beiden Lösungen.

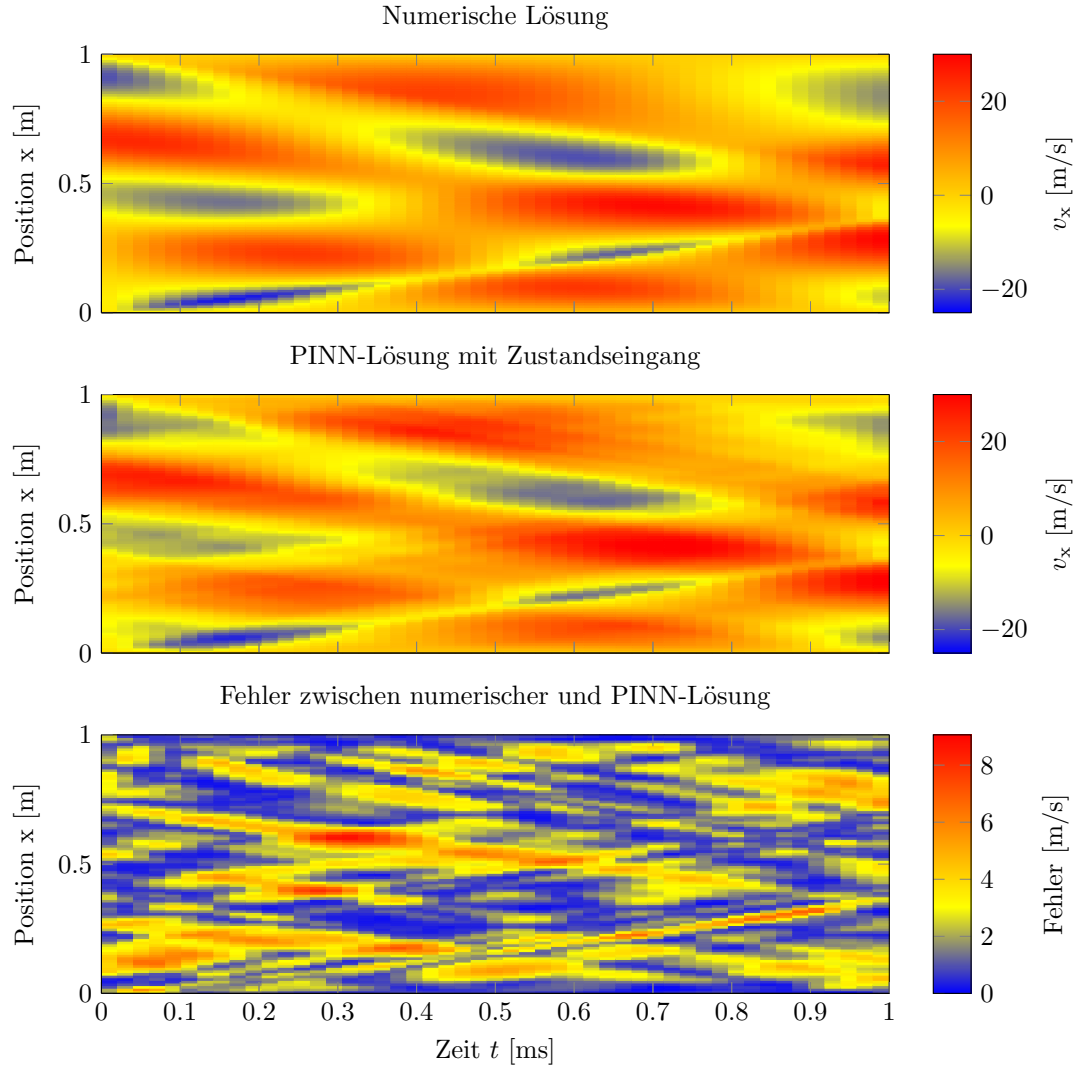


Abbildung 3.11: Vergleich des Verlaufs der Geschwindigkeit aus einer numerischen Lösung und der PINN-Lösung mit Zustandseingang.

Oben: Numerische Lösung mit einer FVM gemäß der Beschreibung in Abschnitt 3.1.4.

Mitte: PINN-Lösung mit Zustandseingang; die Zustandsrepräsentation entspricht den Dichte-, Geschwindigkeits- und Energiewerten an fünf äquidistanten Stützstellen.

Unten: Absoluter Fehler zwischen den beiden Lösungen.

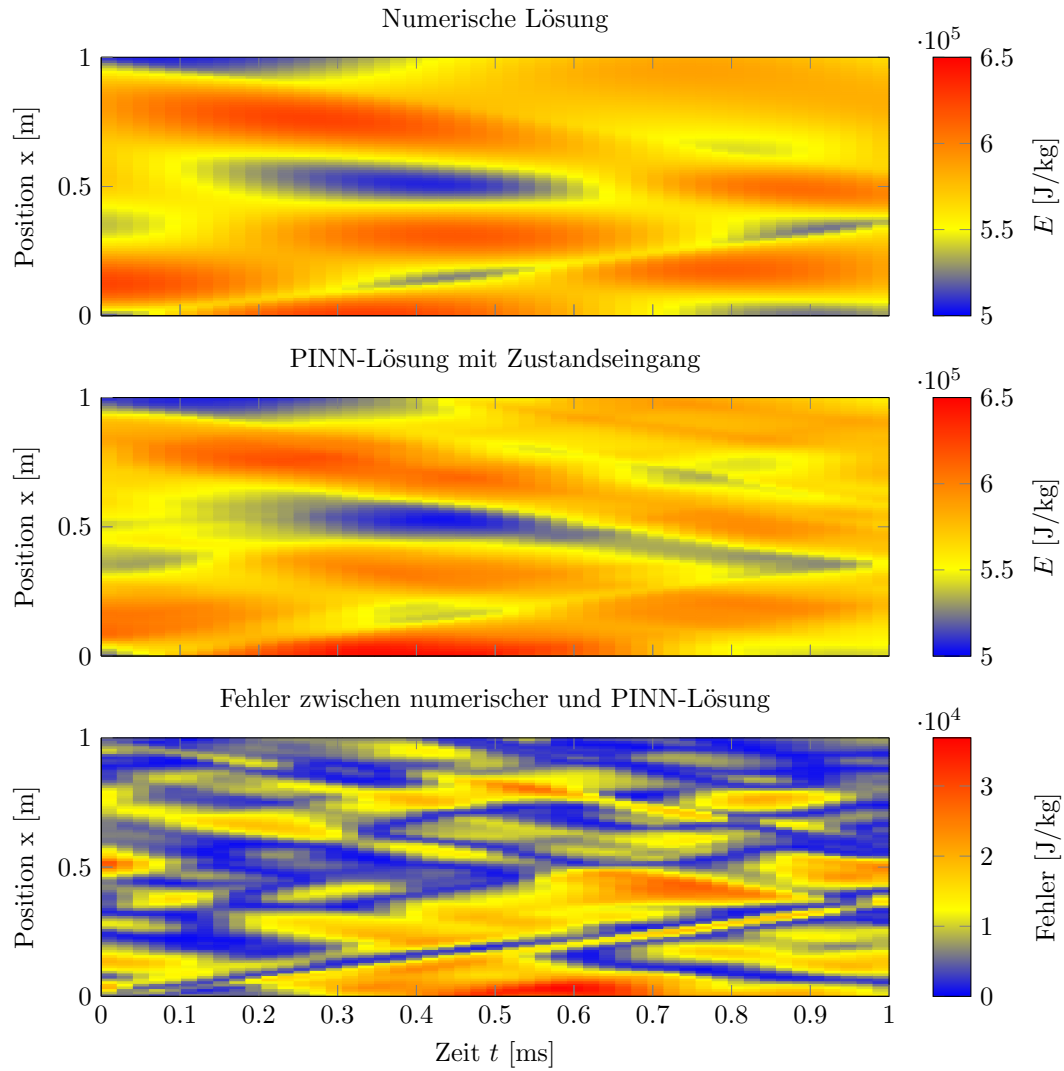


Abbildung 3.12: Vergleich des Verlaufs der Energie aus einer numerischen Lösung und der PINN-Lösung mit Zustandseingang.

Oben: Numerische Lösung mit einer FVM gemäß der Beschreibung in Abschnitt 3.1.4.

Mitte: PINN-Lösung mit Zustandseingang; die Zustandsrepräsentation entspricht den Dichte-, Geschwindigkeits- und Energiewerten an fünf äquidistanten Stützstellen.

Unten: Absoluter Fehler zwischen den beiden Lösungen.

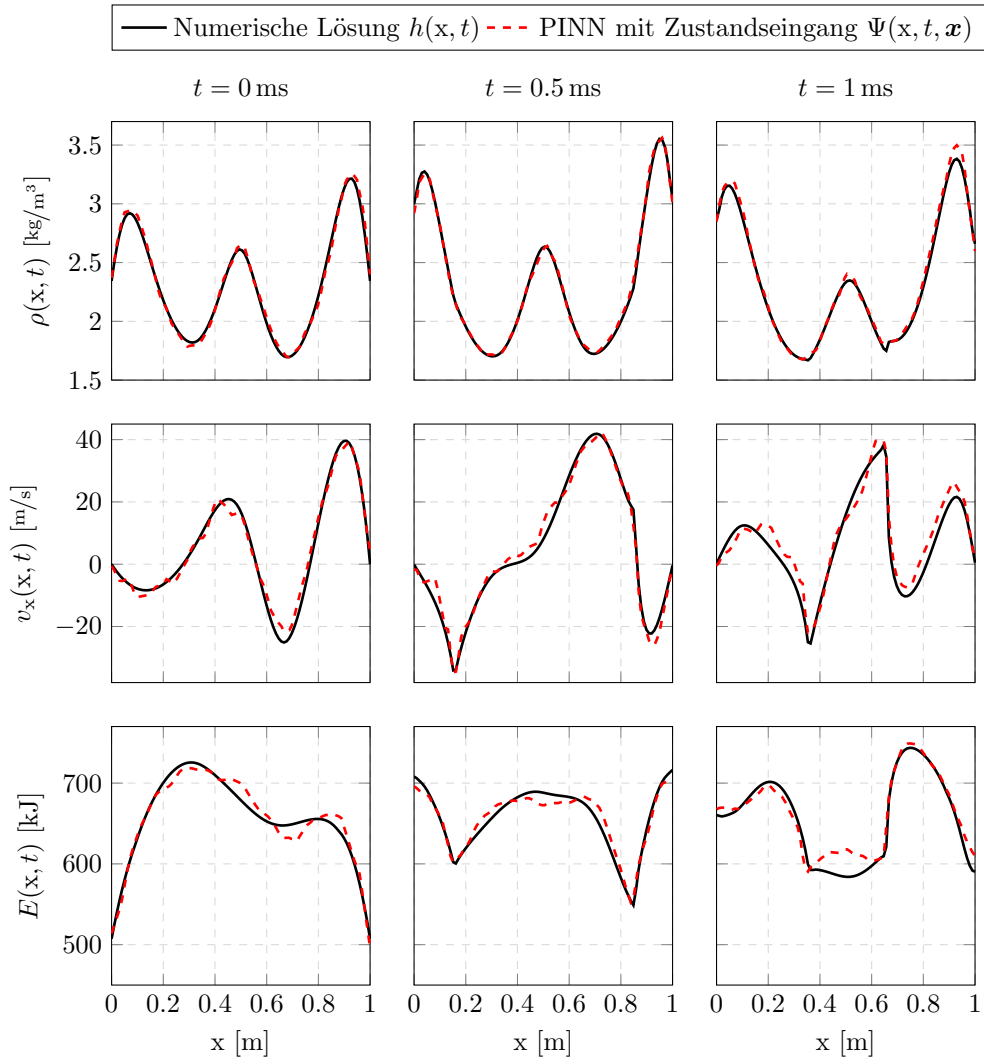


Abbildung 3.13: Vergleich zwischen numerischer Lösung und PINN-Lösung mit Zustandseingang zu den Zeitpunkten $t = 0 \text{ ms}$, $t = 0.5 \text{ ms}$ und $t = 1 \text{ ms}$. Die Zeit $t = 1 \text{ ms}$ entspricht dem maximalen Zeitschritt, der mit dem antrainierten PINN-Modell möglich ist.

4 Zustandsschätzung

In diesem Kapitel wird auf die Verwendung der beschriebenen PINN-Modellierung in einer Zustandsschätzung eingegangen. In vielen technischen Anwendungen ist es nicht möglich, den Systemzustand $\mathbf{x}(t)$ vollständig zu messen. Wird der vollständige Zustand jedoch benötigt, wie es beispielsweise in der modellgestützten Regelung häufig der Fall ist, muss ein Verfahren zu Schätzung der zeitlichen Verläufe der nicht messbaren Größen implementiert werden. Zudem können mit einer Zustandsschätzung Mess- und Modellfehler kompensiert und so die Performance der Regelung verbessert werden. Die Qualität der Zustandsschätzung hängt ebenso wie bei der Regelung vor allem davon ab, wie gut das verwendete Modell das Systemverhalten widerspiegelt.

Im Folgenden wird gezeigt, wie die vorgestellte PINN-Modellierung von dynamischen Systemen auch in der Zustandsschätzung genutzt werden kann, um zuverlässig die Zustände in komplexen Strömungssystemen zu schätzen. Dafür wird auf das weitverbreitete und etablierte Verfahren der erweiterten Kalman-Filterung zurückgegriffen. Nachdem die entsprechende Theorie kurz eingeführt wird, zeigt dieses Kapitel die Umsetzung für jeweils ein System auf Basis der Burgers-Gleichung und anschließend der Euler-Gleichungen.

4.1 Erweitertes Kalman-Filter

Das sogenannte Kalman-Filter wurde erstmals in Kalman, 1960 vorgestellt und hat sich seit seiner Einführung als zuverlässiges Verfahren für diverse Anwendungen, wie modellgestützte Messungen oder Sensorfusion, erwiesen. In dieser Arbeit wird das sogenannte erweiterte Kalman-Filter verwendet, welches den ursprünglichen, für lineare Systeme definierten Ansatz auf den allgemeinen nichtlinearen Fall erweitert (Smith, Schmidt & McGee, 1962; McElhoe, 1966). In den folgenden Abschnitten verweisen Größen mit der $\hat{\cdot}$ Notation darauf, dass es sich um geschätzte Werte handelt.

Ausgangspunkt ist ein allgemeines nichtlineares, diskretes Zustandsraummodell

$$\mathbf{x}_{l+1} = \mathbf{f}(\mathbf{x}_l, \mu_l) + \mathbf{d}_l^f \quad (4.1)$$

$$\gamma_l = \mathbf{z}(\mathbf{x}_l, \mu_l) + \mathbf{d}_l^z \quad (4.2)$$

mit den zufälligen Störungen \mathbf{d}_l^f und \mathbf{d}_l^z , die als Systemrauschen bzw. Messrauschen bezeichnet werden. Beide Störungen werden als weißes Rauschen angenommen. Für die Erwartungswerte gilt folglich

$$\mathcal{E}\{\mathbf{d}_l^f\} = \mathbf{0}, \quad \mathcal{E}\{\mathbf{d}_l^f \mathbf{d}_{l_c}^{f\top}\} = \mathbf{Q} \delta_{Kl, l_c} \quad (4.3)$$

$$\mathcal{E}\{\mathbf{d}_l^z\} = \mathbf{0}, \quad \mathcal{E}\{\mathbf{d}_l^z \mathbf{d}_{l_c}^{z\top}\} = \mathbf{R} \delta_{Kl, l_c} \quad (4.4)$$

mit den Kovarianzmatrizen \mathbf{Q} und \mathbf{R} , welche die Störungen eindeutig charakterisieren.

Das Ziel der Anwendung eines Kalman-Filters ist es, aus den vorhandenen zufälligen Mess- und Modellinformationen, die bis zum Zeitschritt l zur Verfügung stehen, den Wert mit der

4 Zustandsschätzung

kleinsten Schätzfehlerkovarianz für den Zustand $\hat{\mathbf{x}}_{l|l} = \mathcal{E}\{\mathbf{x}_l | \gamma_0, \gamma_1, \dots, \gamma_l\}$ zu schätzen. Dafür werden zunächst Modellinformationen ausgewertet, bevor anschließend die Messinformationen berücksichtigt werden. Dabei erfolgt die Anpassung mit dem Ziel, die Kovarianz des Schätzfehlers \mathbf{P} über die Zeitschritte zu minimieren.

Die einzelnen Schritte des Kalman-Filters für eine Iteration sind in Algorithmus 1 aufgeführt. Eine Herleitung der einzelnen Schritte in Algorithmus 1 ist beispielsweise in Julier

Algorithmus 1 Erweitertes Kalman-Filter

```

1:  $\mathbf{P}_{l|l} = \mathbf{P}_{0|0}$ 
2:  $\mathbf{Q}_l = \mathbf{Q}_0$ 
3:  $\mathbf{R}_{l+1} = \mathbf{R}_0$ 
4:  $\hat{\mathbf{x}}_{l|l} = \hat{\mathbf{x}}_{0|0}$ 
5: while TRUE do
6:    $\triangleright$  Time Update
7:    $\mathbf{P}_{l+1|l} = \mathbf{F}_{l|l} \mathbf{P}_{l|l} \mathbf{F}_{l|l}^T + \mathbf{Q}_l$ 
8:    $\hat{\mathbf{x}}_{l+1|l} = \mathbf{f}(\hat{\mathbf{x}}_{l|l}, \mu_l)$ 
9:    $\hat{\gamma}_{l+1|l} = \mathbf{z}(\hat{\mathbf{x}}_{l+1|l}, \mu_{l+1})$ 
10:   $\triangleright$  Measurement Update
11:   $\mathbf{K}_{l+1} = \mathbf{P}_{l+1|l} \mathbf{Z}_{l+1|l}^T [\mathbf{Z}_{l+1|l} \mathbf{P}_{l+1|l} \mathbf{Z}_{l+1|l}^T + \mathbf{R}_{l+1}]^{-1}$ 
12:   $\hat{\mathbf{x}}_{l+1|l+1} = \hat{\mathbf{x}}_{l+1|l} + \mathbf{K}_{l+1} (\gamma_{l+1} - \hat{\gamma}_{l+1|l})$ 
13:   $\mathbf{P}_{l+1|l+1} = \mathbf{P}_{l+1|l} - \mathbf{K}_{l+1} \mathbf{Z}_{l+1|l} \mathbf{P}_{l+1|l}$ 
14: end while

```

und Uhlmann, 2004 gegeben. Für die Initialisierung der Matrizen \mathbf{P} , \mathbf{Q} und \mathbf{R} gibt es keine einheitlichen Regeln. Das Verhalten des Kalman-Filters wird jedoch maßgeblich von der Anpassung dieser Parameter bestimmt. Einige allgemeine Empfehlungen zu diesem Thema sind in Schneider und Georgakis, 2013 und Neddermeyer und King, 2019 zu finden.

Ein weiterer entscheidender Faktor für die Qualität der Zustandsschätzung ist die Genauigkeit, mit der das System linearisiert wird. Besteht die Möglichkeit, für die Jakobi-Matrizen

$$\mathbf{F}_{l|l} = \left. \frac{\partial \mathbf{f}}{\partial \mathbf{x}} \right|_{\mathbf{x}=\hat{\mathbf{x}}_{l|l}} \quad \text{und} \quad \mathbf{Z}_{l+1|l} = \left. \frac{\partial \mathbf{z}}{\partial \mathbf{x}} \right|_{\mathbf{x}=\hat{\mathbf{x}}_{l+1|l}} \quad (4.5)$$

eine analytische Ableitung aus den Zustands- und Messgleichung zu berechnen, verspricht dieser Ansatz die besten Ergebnisse. Zudem ist die Auswertung einer analytischen Ableitung zumeist sehr recheneffizient möglich. In Fällen, in denen eine analytische Ableitung nicht möglich ist, ist eine Bestimmung über eine numerische Methode, z.B. mittels Approximation über ein zentrales Differenzenschema, möglich. In diesem Fall hängt die Genauigkeit der Ableitung stark von der gewählten Schrittweite des numerischen Verfahrens ab. Außerdem erfordert die numerische Berechnung eine häufige Auswertung der Funktion, was den Rechenaufwand erhöht.

Für das vorgestellte PINN-Zustandsraummodell ist eine analytische Ableitung nicht möglich. Dies liegt zum einen an der Komplexität des Modells, vor allem aber an den mitunter nicht vollständig differenzierbaren Aktivierungsfunktionen, die in den Berechnungen der Neuronen enthalten sind. Alternativ bietet sich für die Differenzierung das Verfahren der automatischen Differenzierung an, welches bereits im Abschnitt zum Training der PINN-Modelle erwähnt

wurde. Da der Algorithmus für die automatische Differenzierung allerdings im Gegensatz zu einer analytischen Lösung lediglich die Ableitung an einer bestimmten Stelle liefert, hängt es von der effizienten Umsetzung des Algorithmus auf der entsprechenden Hardware ab, ob sich ein Rechenzeitvorteil gegenüber einer numerischen Ableitung ergibt. Sollte eine automatische Differenzierung umsetzbar sein, ist sie aufgrund der Genauigkeit dem numerischen Ansatz vorzuziehen.

4.2 Erweitertes Kalman-Filter für die Burgers-Gleichung

Um die Ergebnisse möglichst einfach und verständlich zu präsentieren, wird im Folgenden ein autonomes System auf Basis der Burgers-Gleichung betrachtet. Die Aktuation wird entsprechend über den gesamten Zeithorizont als konstant $\mu(t) = 0$ angenommen. Als System, welches zu beobachten ist, wird die numerische Lösung der Burgers-Gleichung auf Basis einer FVM verwendet. Ausgangspunkt ist wie im Beispiel aus Abschnitt 3.3 die Anfangsbedingung

$$h(x, 0) = -\sin(\pi x) \quad (4.6)$$

in einem Bereich $x \in [-1; 1]$. Zur Beschreibung des Systems wird ein PINN-Modell entsprechend Abschnitt 3.3 mit zehn Zuständen als Eingang, die zehn räumlich äquidistant verteilten Werten des Geschwindigkeitsfelds entsprechen, aufgebaut.

Für eine erste Betrachtung wird angenommen, dass die Zustände x_2, x_5, x_6, x_9 im System messbar sind.¹ Es ergibt sich folglich die Messgleichung

$$\mathbf{y}_l = \mathbf{z}(\mathbf{x}_l, 0) = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix} \cdot \mathbf{x}_l \quad (4.7)$$

Für die Abtastrate des System wird eine Frequenz von $f_y = 10$ Hz angenommen. Abbildung 4.1 zeigt einen Intensitätsplot des zu beobachtenden Systems in der Raum-Zeit-Ebene. Die

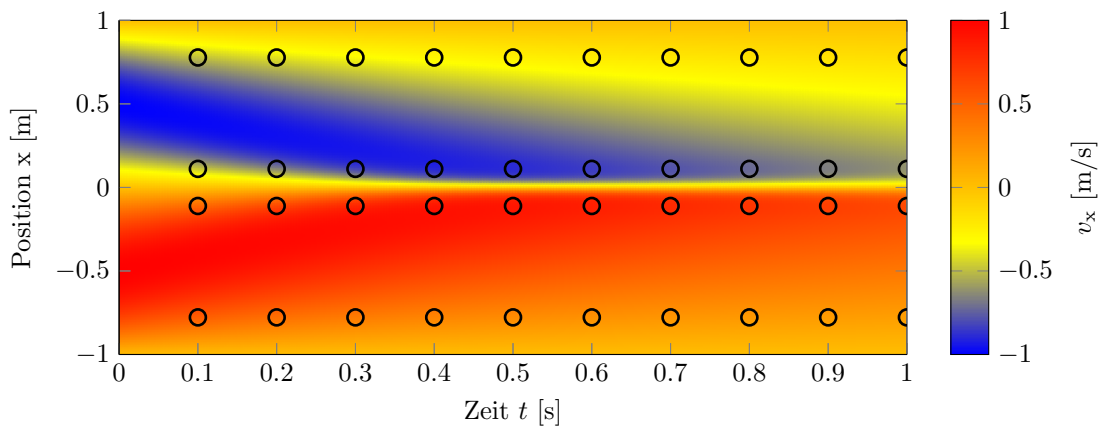


Abbildung 4.1: Intensitätsplot in der Raum-Zeit-Ebene des Systems, dessen Zustand geschätzt werden soll. Die Messpunkte sind mit schwarzen Kreisen markiert. Das System wird durch die numerische Lösung der Burgers-Gleichung definiert.

¹Grundsätzlich sind auch andere Zustandskombinationen für die Messung möglich, allerdings ist das System je nach Wahl nicht mehr vollständig aus der Messung beobachtbar.

4 Zustandsschätzung

Punkte in der Raum-Zeit-Ebene, für die Messungen zur Verfügung stehen, sind entsprechend markiert.

Die Parameter des Kalman-Filters wurden so eingestellt, dass sich das gewünschte Verhalten in dem dargestellten Zeithorizont $t \in [0, 1]$ gut erkennen lässt. Die initiale Kovarianz des Schätzfehlers und des Systemrauschens werden auf

$$\mathbf{P}_{0|0} = 0.5 \cdot \mathbf{I}^{10} \quad \mathbf{Q}_0 = 0.5 \cdot \mathbf{I}^{10} \quad (4.8)$$

festgelegt. Hier beschreibt \mathbf{I}^{10} die Einheitsmatrix der Dimension zehn. Für das Messrauschen wird ebenfalls die Einheitsmatrix

$$\mathbf{R}_0 = \mathbf{I}^{10} \quad (4.9)$$

angenommen. In mehreren Versuchen zeigte sich, dass der Unterschied zwischen der Berechnung der Ableitung mit numerischen Methoden und mittels automatischem Differenzieren für dieses Beispiel sehr gering ist. Aufgrund der in Abschnitt 4.1 beschriebenen Vorteile, werden die Ableitungen im Folgenden immer mittels automatischem Differenzieren bestimmt.

Um zu verdeutlichen, wie gut das PINN-Modell in diesem beispielhaften Szenario einer Zustandsschätzung funktioniert, wird zunächst ein Extremfall betrachtet, in dem angenommen wird, dass keinerlei Information über die Anfangsbedingung vorliegt. Daher wird der geschätzte Anfangszustand als $\hat{\mathbf{x}}_0 = \mathbf{0}$ gewählt. Abbildung 4.2 zeigt auf der linken Seite die zeitliche Entwicklung der messbaren Zustände und deren Schätzungen. Es ist zu erkennen, dass die

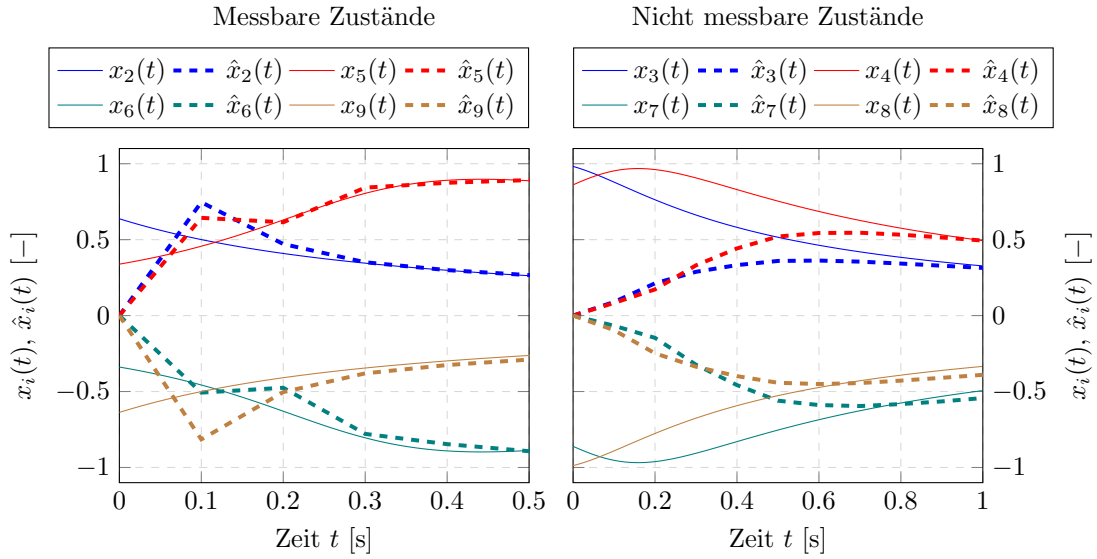


Abbildung 4.2: Zeitliche Verläufe der messbaren bzw. nicht messbaren Zustände und der entsprechenden Schätzungen, die durch das EKF berechnet werden. Aufgrund der langsameren Dynamik sind die nicht messbaren Zustände über einen längeren Zeitraum aufgetragen.

messbaren Zustände bereits zum Zeitpunkt $t = 0.3$ s, d.h. nach dem dritten *Measurement Update*, eine gute Übereinstimmung zeigen. Zum Zeitpunkt $t = 0.5$ s stimmen die Schätzungen für die messbaren Zustände annähernd mit den tatsächlichen Werten überein. Eine noch aggressivere Abstimmung des Kalman-Filters mit einer kleineren Kovarianz des Messrauschens \mathbf{R} führt zu einer noch schnelleren Übereinstimmung von Messung und Schätzung. Dies wurde hier jedoch nicht gewählt, um den Bezug zu einer realen Anwendung herzustellen, in der mit einem größeren Messrauschen zu rechnen ist.

4.2 Erweitertes Kalman-Filter für die Burgers-Gleichung

Das für die Qualität der Schätzung besonders relevante Ergebnis ist in Abb. 4.2 auf der rechten Seite zu finden. Hier sind die zeitlichen Verläufe der nicht messbaren Zustände x_3, x_4, x_7, x_8 und die entsprechenden Schätzungen dargestellt.² Zwar ist hier eine genaue Schätzung erst zum Zeitpunkt $t = 1$ s erreicht, jedoch ist die Qualität der Schätzung ab diesem Zeitpunkt ähnlich zu den Ergebnissen der messbaren Zustände. Insgesamt zeigt dieses Ergebnis, dass über den PINN-Ansatz ein Modell aufgebaut werden kann, welches auch mit relativ wenigen Messinformationen eine gute Schätzung des Systemzustands ermöglicht.

Um den Vorteil einer Zustandsschätzung gegenüber der reinen Modellprädiktion zu verdeutlichen, ist in Abb. 4.3 ein direkter Vergleich dargestellt. Dafür wurde abweichend von dem

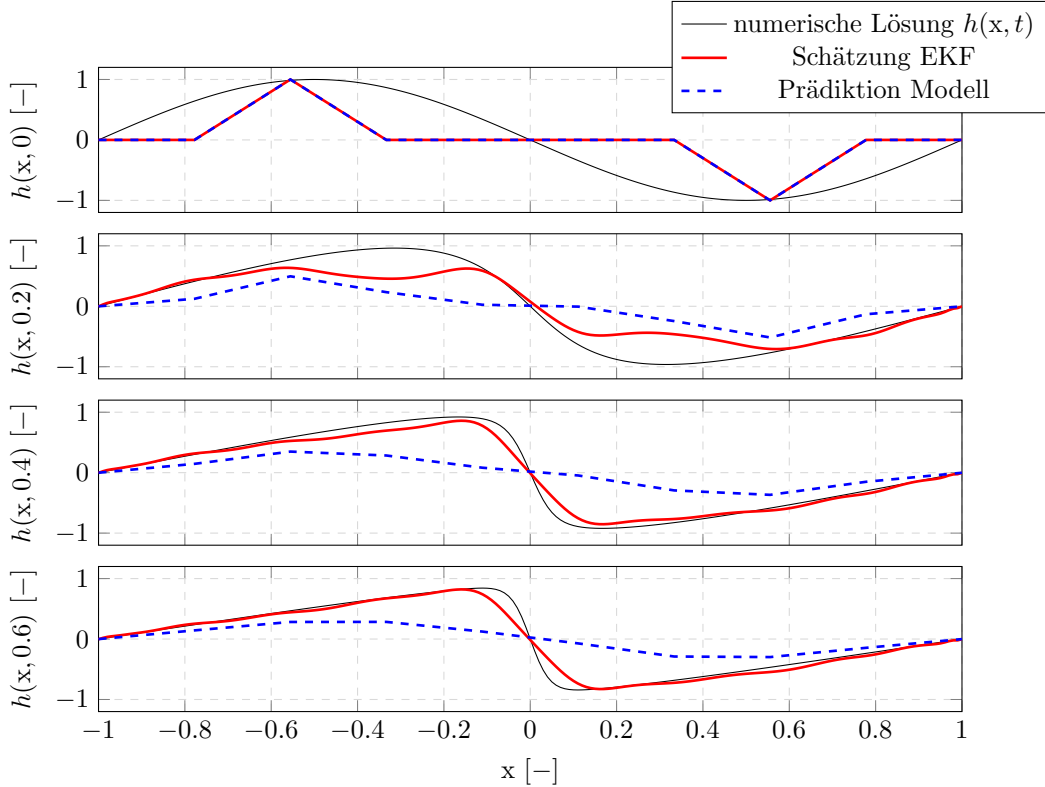


Abbildung 4.3: Vergleich der numerischen Lösung der Burgers-Gleichung mit einer Schätzung mit einem EKF auf Basis des PINN-Modells und einer Prädiktion ausschließlich mit dem PINN-Modell. Für die Interpolation zwischen den Zuständen wurde dabei das PINN-Modell selbst benutzt.

vorangegangenen Beispiel eine anfängliche Schätzung des Zustandes

$$\hat{\mathbf{x}}_0 = [0, 0, 1, 0, 0, 0, 0, 1, 0, 0]^T \quad (4.10)$$

angenommen. Für die Darstellung in Abb. 4.3 wurde im Kalman-Filter der geschätzte Zustand genutzt, um das PINN-Modell an 200 räumlich äquidistanten Punkten auszuwerten und die gesamte räumliche Information zu erhalten. Für die Modellprädiktion wurde das PINN-Modell in jedem Zeitschritt an 200 räumlich äquidistanten Punkten ausgewertet. Eine Korrektur des Zustandes durch Messinformationen erfolgte nicht. Es ist zu erkennen, dass aufgrund der sehr ungenauen Schätzung des Anfangszustandes die Modellprädiktion in keinem Zeitschritt das Systemverhalten zufriedenstellend abbildet. Im Gegensatz dazu verbessert sich die Schätzung mit dem Kalman-Filter deutlich über die Zeit. Zum Zeitpunkt $t = 0.6$ s

²Die Zustände x_1 und x_{10} sind nicht dargestellt, da sie den bekannten Randbedingungen entsprechen.

zeigt die Schätzung eine gute Übereinstimmung mit der realen Geschwindigkeitsverteilung. Lediglich an Stellen mit starken Geschwindigkeitsänderungen sind leichte Fehler der Schätzung zu erkennen. Dieses Ergebnis könnte durch die Berücksichtigung zusätzlicher Zustände im PINN-Modell möglicherweise noch verbessert werden. In den Untersuchungen zeigte jedoch die gewählte Zustandsgröße mit der Dimension zehn den besten Kompromiss zwischen Rechenaufwand, Modellgenauigkeit und Generalisierung.

4.3 Erweitertes Kalman-Filter für die Euler-Gleichungen

Um zu zeigen, wie gut der PINN-Modellierungsansatz auch sehr komplexe Systeme abbildet, wird im Folgenden auf ein Beispiel einer Zustandsschätzung für ein System eingegangen, welches mit den Euler-Gleichungen beschrieben werden kann. Wie in Abschnitt 3.1.2 beschrieben, können die Euler-Gleichungen genutzt werden, um das Verhalten realer Strömungsprozesse in guter Näherung abzubilden. Das zu beobachtende System wird auch hier von der in Abschnitt 3.1.4 beschriebenen numerischen 1-D-Simulation der Euler-Gleichungen abgebildet. Das simulierte System ist in diesem Fall ein konstant durchströmtes Rohr, an dessen Auslass Umgebungsbedingungen herrschen. Am Einlass wird ein variabler Druck zwischen $p = 1$ bar und $p = 3$ bar simuliert. Dabei wird angenommen, dass alle anderen Zustandsgrößen wie die Temperatur ϑ und Geschwindigkeit v_x am Einlass konstant sind.³ Zusätzlich wird eine begrenzte Druckänderungsrate am Einlass von 1 bar/ms angenommen.⁴ Die wichtigsten Kenngrößen der als System angenommenen numerischen Simulation sind in Abbildung 4.4 nochmal visualisiert.

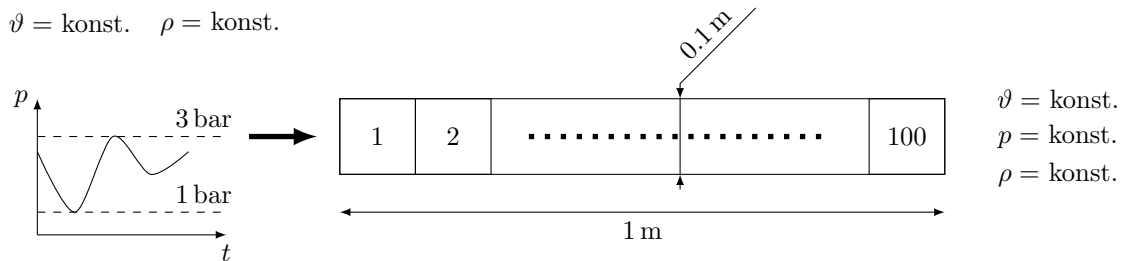


Abbildung 4.4: Aufbau der Simulation auf Basis der Euler-Gleichungen, die als System in der Zustandsschätzung verwendet wird.

Wie für das in Abschnitt 3.4.2 diskutierte Euler-System mit reflektierenden Randbedingungen, ist auch für dieses Beispiel ein zweistufiges Training von drei separaten Netzen Ψ_ρ , Ψ_{v_x} und Ψ_E für die Approximation von Dichte, Geschwindigkeit und Energie notwendig. Abweichend zu dem in Abschnitt 3.4.2 beschriebenen Beispiel, wird für den aktuierten Fall eine Zustandsbeschreibung aus zehn äquidistant verteilten Werten für Dichte, Geschwindigkeit und Energie gewählt. Entsprechend vergrößert sich die Dimension des Zustandes auf $n_x = 30$ im Vergleich zu $n_x = 15$ im Beispiel mit reflektierenden Randbedingungen. Die Aktuation, sprich die Veränderung des Drucks am Rohreinlass, wird in diskrete Schritte unterteilt. Die

³Diese Annahmen bildet zwar nicht die Realität ab, sollen hier aber eine einfache Nachvollziehbarkeit und Darstellung der Ergebnisse ermöglichen.

⁴Eine derart schnelle Änderung der Einlassbedingungen ist technisch wahrscheinlich nicht zu realisieren, es soll jedoch in diesem Beispiel die hohe Dynamik verdeutlichen, die mit einem PINN-Zustandsraummodell zufriedenstellend abgebildet werden kann.

Menge der diskreten Aktuationsschritte

$$\mathcal{M} = \{1.0 \text{ bar}, 1.1 \text{ bar}, \dots, 2.9 \text{ bar}, 3.0 \text{ bar}\} \quad (4.11)$$

besteht hier aus $n_{\mathcal{M}} = 21$ Elementen. Für jeden dieser Aktuationsschritte müssen entsprechend den Ausführungen in Abschnitt 3.4.2 die drei Netze Ψ_{ρ} , Ψ_{v_x} und Ψ_E antrainiert werden, um dann über die Interpolation das entsprechende Zustandsraummodell zu bilden. Die Netze Ψ_{ρ} , Ψ_{v_x} und Ψ_E bestehen jeweils aus acht Schichten mit 100 Neuronen sowie einer Eingangsschicht mit 32 Neuronen und einem Ausgangsneuron. Als Aktivierungsfunktion wird die ReLu-Funktion verwendet.

Die Auswahl einer sinnvollen Trainingsmenge ist für die Euler-Gleichungen im Vergleich zur Burgers-Gleichung deutlich schwieriger. Für die Burgers-Gleichung konnte mit komplett zufälligen Anfangsbedingungen ein sehr robustes Netz antrainiert werden, welches für eine Vielzahl von Anfangsbedingungen gute Approximationen liefert. Umfangreiche Versuche für die Euler-Gleichungen haben gezeigt, dass mit einer ähnlichen Zusammensetzung der Trainingsmenge nur unzureichend genaue Approximationen erzielt werden können. Dies ist vor allem auf die höhere Komplexität des partiellen Differentialgleichungssystems zurückzuführen, die es erlaubt, ein breiteres Spektrum von verschiedenen Strömungsphänomenen abzubilden. Für bessere Ergebnisse muss daher die Trainingsmenge passend für das zu beschreibende System gewählt werden. Dafür müssen die Daten so erzeugt werden, dass im Training nur Anfangsbedingungen berücksichtigt werden, die auch in der Auswertung des Modells in ähnlicher Form auftreten können. Dies gilt vor allem für Größenordnungen und Frequenzen. Daher wurden für dieses Beispiel keine komplett zufälligen Anfangsbedingungen, sondern Trainingsdaten aus einem fortlaufenden Datensatz ausgewählt, der aus einer Anregung des Systems mit einem zufälligen Eingangssignal entstanden ist. Das Eingangssignal, also der zeitliche Verlauf des Drucks am Einlass des Rohres ist in Abb. 4.5 dargestellt. Dabei wurde auch die Beschränkung

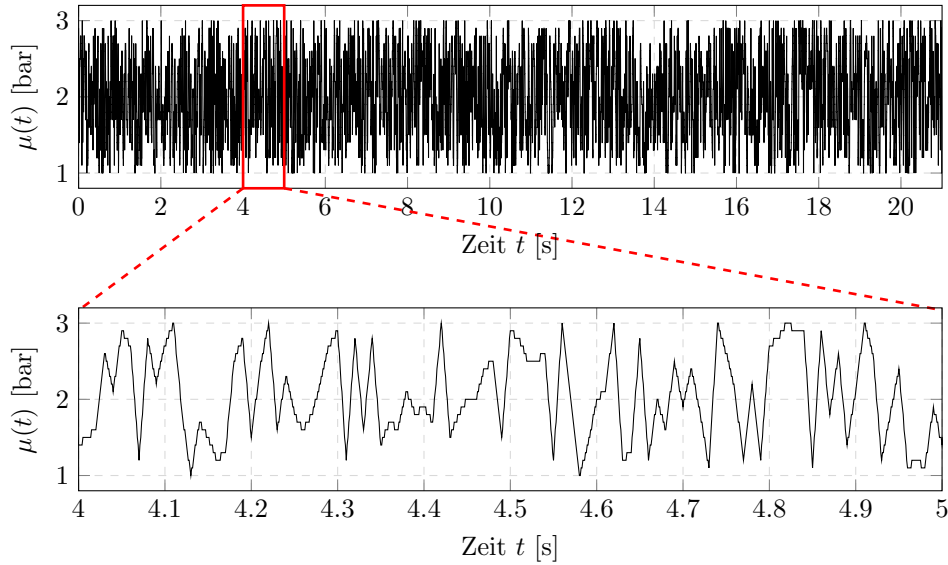


Abbildung 4.5: Anregungssignal des numerischen Systems für die Gewinnung der Trainingsdaten.

der Eingangsdynamik von 1 bar/ms berücksichtigt. Da aufgrund der Charakteristik des Anregungssignals die Aktuationsschritte nicht gleichverteilt über der Zeit auftreten, entstehen unterschiedlich große Trainingsmengen für die jeweiligen Aktuationsschritte. Im Speziellen nimmt die Größe der Trainingsmenge zu den Extremwerten $\mu = 1.0 \text{ bar}$ und $\mu = 3.0 \text{ bar}$ ab. Eine Verteilung der Größen der Trainingsmengen über die Aktuationsschritte ist in Abb. 4.6

4 Zustandsschätzung

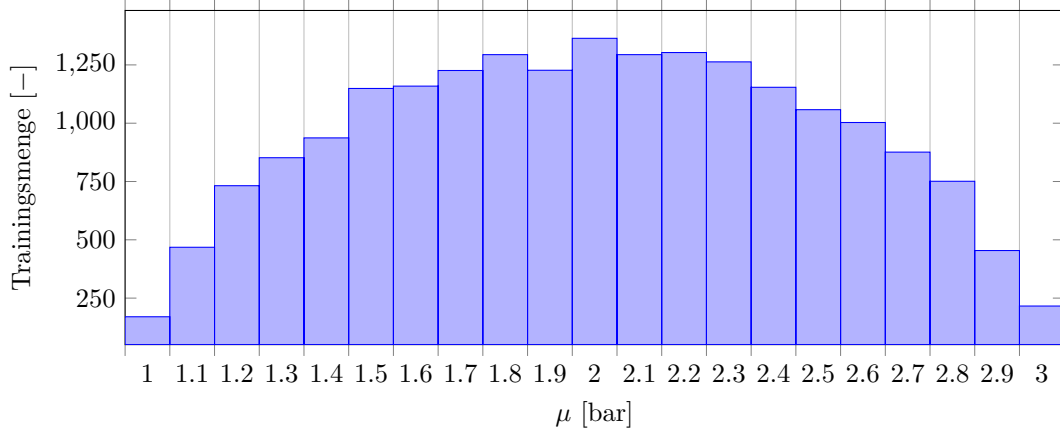


Abbildung 4.6: Verteilung der Größe der Trainingsmengen über die Aktuierungsschritte. Für jede Anfangsbedingung wurden in der ersten und zweiten Stufe des Trainings 300 Rand- bzw. Anfangspunkte und 5000 Kollokationspunkte berücksichtigt.

zu finden. Die Länge des Eingangssignals wurde so gewählt, dass auch für die kleinsten Trainingsmengen im Training ein zufriedenstellendes Ergebnis erzielt werden kann. Für die Validierung des Trainingsergebnisses wurden jeweils 10 % der Daten berücksichtigt. Die Netze wurden für eine maximale Zeitschrittweite von $\Delta t = 1$ ms antrainiert.

Wie beschrieben, wird der Zustand des Systems durch jeweils zehn Werte für Dichte, Strömungsgeschwindigkeit und Energie repräsentiert. Für eine Zustandsschätzung ist diese Zusammensetzung des Zustandes problematisch, da eine Messung einzelner Zustände in der Praxis schwer zu realisieren ist. Die Messung der Dichte und vor allem der spezifischen Energie ist technisch nur bedingt umsetzbar. Auch die Messung der Geschwindigkeit kann je nach Position technisch schwierig oder unmöglich sein. In der Praxis sind vor allem Druck- und Temperaturmessungen einfach zu realisieren. Auch wenn das hier betrachtete System vor allem als akademisches Beispiel zur Veranschaulichung des Konzeptes zu sehen ist, wird versucht, die Messgrößen für die Zustandsschätzung möglichst praktikabel für eine reale Umsetzung zu wählen. Es wird angenommen, dass an zehn äquidistanten Messpositionen der Druck und die Temperatur gemessen werden. Zudem wird angenommen, dass die Strömungsgeschwindigkeit am Ein- und Auslass gemessen werden kann. Unter der Annahme der idealen Gasgleichung mit einer spezifischen Gaskonstanten $R = 287.1 \text{ J/kg K}$ und einem konstanten Isentropenexponenten von $\gamma = 1.4$ ergibt sich damit die Messgleichung

$$\mathbf{y}_l = \begin{bmatrix} \vartheta_{l,1} \\ \vdots \\ \vartheta_{l,10} \\ p_{l,1} \\ \vdots \\ p_{l,10} \\ v_{l,\text{Ein}} \\ v_{l,\text{Aus}} \end{bmatrix} = \begin{bmatrix} \frac{0.4}{R} \left(E_{l,1} - \frac{1}{2} v_{l,1}^2 \right) \\ \vdots \\ \frac{0.4}{R} \left(E_{l,10} - \frac{1}{2} v_{l,10}^2 \right) \\ 0.4 \cdot \rho_{l,1} \left(E_{l,1} - \frac{1}{2} v_{l,1}^2 \right) \\ \vdots \\ 0.4 \cdot \rho_{l,10} \left(E_{l,10} - \frac{1}{2} v_{l,10}^2 \right) \\ v_{l,1} \\ v_{l,10} \end{bmatrix} = \begin{bmatrix} \frac{0.4}{R} \left(x_{l,3} - \frac{1}{2} x_{l,2}^2 \right) \\ \vdots \\ \frac{0.4}{R} \left(x_{l,30} - \frac{1}{2} x_{l,29}^2 \right) \\ 0.4 \cdot x_{l,1} \left(x_{l,3} - \frac{1}{2} x_{l,2}^2 \right) \\ \vdots \\ 0.4 \cdot x_{l,28} \left(x_{l,30} - \frac{1}{2} x_{l,29}^2 \right) \\ x_{l,2} \\ x_{l,29} \end{bmatrix}. \quad (4.12)$$

Diese Messgleichung ist durch die als konstant angenommenen Werte für Gaskonstante und Isentropenexponent nur eine Approximation, die in einem bestimmten Bereich eine gute Näherung ermöglicht. In dem hier betrachteten Beispiel ist die Variation der beiden Werte sehr

4.3 Erweitertes Kalman-Filter für die Euler-Gleichungen

gering. Zudem berücksichtigt der Ansatz des EKF's wie oben beschrieben auch eine Unsicherheit in der Messung, die kompensiert werden kann. Für ein System mit einem breiteren Betriebsbereich ist jedoch mitunter eine Anpassung der Messgleichung notwendig.

Für die Anfangsbedingung wird eine konstante Geschwindigkeit und Temperatur

$$v_x(0, x) = 150 \frac{\text{m}}{\text{s}}, \quad \vartheta(0, x) = 300 \text{ K} \quad (4.13)$$

sowie eine sinusförmige Druckverteilung

$$p(0, x) = [2 + 0,5 \cdot \sin(2\pi \cdot x)] \text{ bar} \quad (4.14)$$

angenommen.⁵ Um zu verdeutlichen, wie gut das Modell in einer Zustandsschätzung funktionieren kann, wird in diesem Beispiel davon ausgegangen, dass der initiale Zustand für die Schätzung vollkommen unbekannt ist. Daher wird der geschätzte Zustand mit

$$\hat{x}_0 = \mathbf{0} \quad (4.15)$$

initialisiert. Der Zeitschritt für das Zeit- und Mess-Update wird auf die maximal durch das Modell abbildbare Zeitschrittweite von $\Delta t = 1 \text{ ms}$ festgelegt. Die Parameter des erweiterten Kalman-Filters werden mit

$$\mathbf{P} = \mathbf{I}^{30}, \quad \mathbf{Q} = \mathbf{I}^{30}, \quad \mathbf{R} = \mathbf{I}^{22} \quad (4.16)$$

initialisiert. Diese Einstellung ist nur möglich, da die gesamte Rechnung mit normierten Werten erfolgt. Andernfalls müssten die Parameter an die entsprechende Größenordnung der Zustände und Messungen angepasst werden. Für die Anregung des Systems $\mu(t)$ wird in diesem Beispiel ein wertediskretes Sägezahnsignal zwischen 1 bar und 3 bar mit einer Frequenz von 2.5 Hz angenommen. Das Eingangssignal ist in Abb. 4.7 unten dargestellt. Für einen ersten Eindruck der Schätzung ist in Abb. 4.7 der mittlere absolute Schätzfehler der Zustände abgebildet. Diese beschreiben die Dichte, Geschwindigkeit und Energie. Zusammen mit der ebenfalls dargestellten Standardabweichung zeigt sich, dass sich die Schätzung der Zustände innerhalb der ersten 20 ms deutlich verbessert. Danach verschlechtert sich die Schätzung wieder merklich. Dies ist auf die Bedingungen vor und hinter dem Rohr zurückzuführen. Im ersten Teil ist die Druckdifferenz zwischen Ein- und Auslass relativ groß, was zu eindeutigeren Strömungsbedingungen führt, die mit dem Modell sehr gut beschrieben werden können. Im zweiten Teil des betrachteten Zeitraums ist die Druckdifferenz zwischen Ein- und Auslass geringer was dazu führt, dass die Auswirkungen der Aktuation länger im System zu sehen sind und das Modell dieses Verhalten schlechter abbildet. Insgesamt ist jedoch die Qualität der Zustandsschätzung als zufriedenstellend anzusehen, da die dominierenden Charakteristiken des Systems gut geschätzt werden können.

Um dies zu verdeutlichen, sind in den Abb. 4.8, 4.9 und 4.10 die Intensitätsplots der numerischen Lösung und der Schätzung des EKF für den Zeitraum $t \in [23; 25] \text{ ms}$ dargestellt. Dabei wurde das PINN-Modell selbst verwendet, um aus den geschätzten Zuständen die Werte auf einem Raum-Zeit-Gitter mit 200×41 äquidistanten Stützstellen zu berechnen. Die numerische Lösung wurde auf das gleiche Gitter interpoliert. Es ist zu erkennen, dass die Schätzung das Verhalten des Systems für alle drei Größen in guter Näherung wiedergibt. Die dominierenden Charakteristiken, die in der numerischen Lösung zu erkennen sind, finden sich ebenfalls gut erkennbar in der Schätzung wieder.

⁵Die hier durch Temperatur, Geschwindigkeit und Druck definierte Anfangsbedingung ergibt unter Zuhilfenahme der thermodynamischen Zustandsgleichungen die eindeutigen initialen Verläufe von Dichte, Geschwindigkeit und spezifischer Energie.

4 Zustandsschätzung

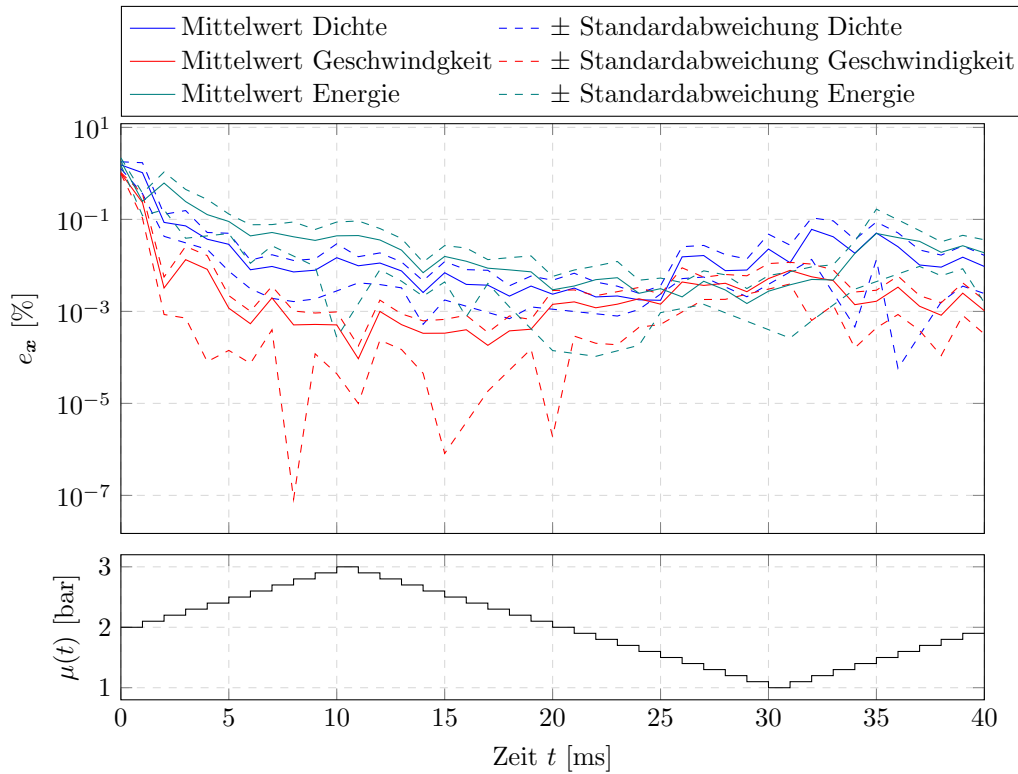


Abbildung 4.7: Oben: Mittlerer relativer Schätzfehler zusammen mit zugehöriger Standardabweichung (gestrichelt) der Zustände, die Dichte, Geschwindigkeit und Energie repräsentieren.

Unten: Wertediskretes Sägezahnsignal, mit dem das System angeregt wurde.

Neben dem guten Schätzergebnis ist vor allem der Rechenaufwand der Zustandsschätzung anzumerken. Im Mittel werden für die Berechnung der Zustandsschätzung in einem Zeitschritt etwa 15 ms benötigt. Zwar liegt dies deutlich über der Zeitschrittweite von $\Delta t = 1$ ms, allerdings wurde die Berechnung auf einem Heimrechner mit einem 4x4GHz i7-Prozessor des Modelljahres 2015 ohne weitere Hardwarebeschleuniger durchgeführt. Für den Aufwand der Berechnung der Ableitungen zeigte sich auf diesem System annähernd kein Unterschied zwischen numerischer Lösung und automatischem Differenzieren. Es ist jedoch davon auszugehen, dass durch eine optimierte Implementierung in einem System mit Hardwarebeschleunigung, z.B. durch GPUs oder FPGAs, der Rechenaufwand nochmal deutlich reduziert werden kann. Dies gilt insbesondere, da die Auswertung des Modells hauptsächlich Matrixmultiplikationen erfordert und damit sehr gut parallelisierbar ist. Insgesamt ist zu erwarten, dass bei passender Implementierung der Rechenaufwand soweit reduziert werden kann, dass eine Umsetzung der Zustandsschätzung in Echtzeit möglich ist. Dies unterscheidet den Ansatz deutlich von bisherigen Ansätzen zur Zustandsschätzung für Systeme auf Basis der Euler-Gleichungen, wie z.B. dem in Schäpel et al., 2017 beschriebenen Ansatz mit einer konventionellen Modellreduktion.

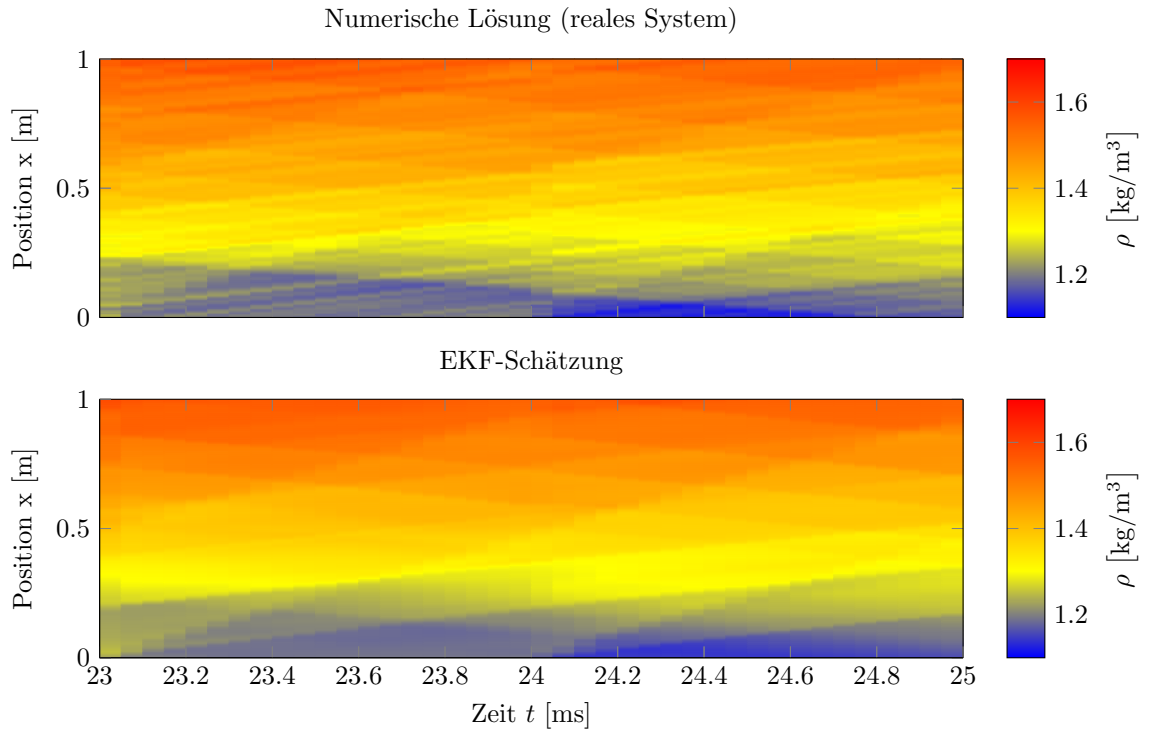


Abbildung 4.8: Intensitätsplot der numerischen Lösung und Schätzung der Dichte. Für die Schätzung wurde das PINN-Modell ausgehend von den geschätzten Zuständen ausgewertet.

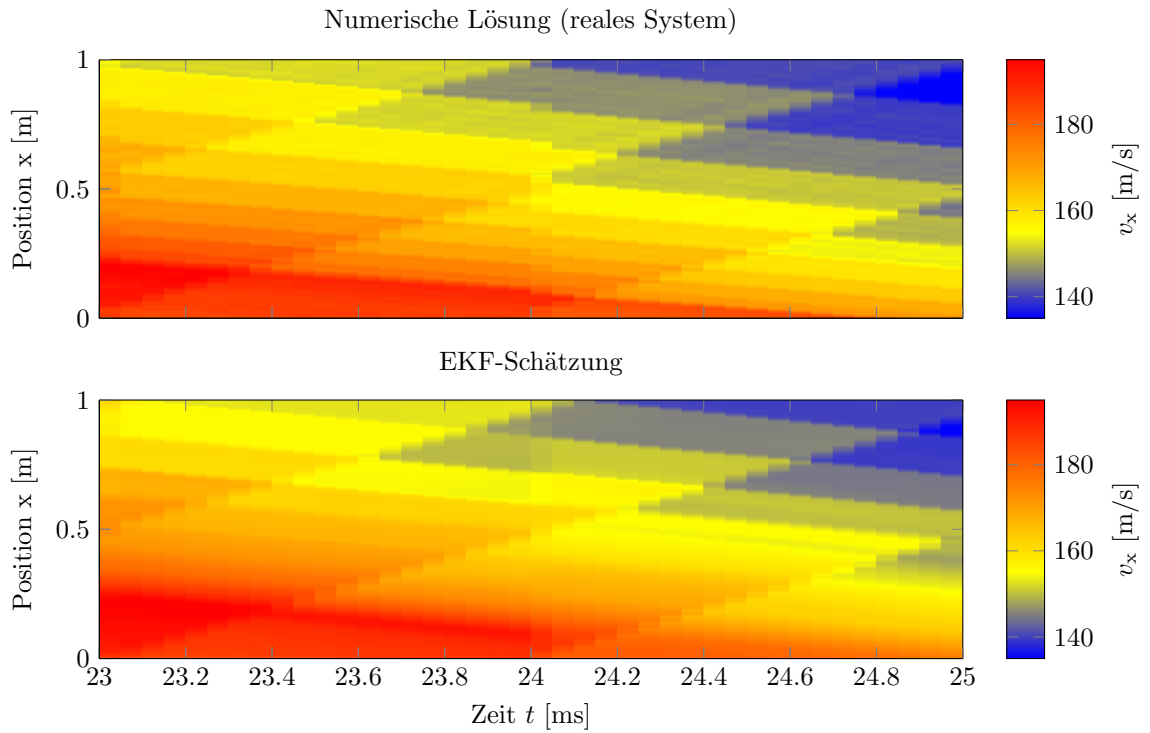


Abbildung 4.9: Intensitätsplot der numerischen Lösung und Schätzung der Strömungsgeschwindigkeit. Für die Schätzung wurde das PINN-Modell ausgehend von den geschätzten Zuständen ausgewertet.

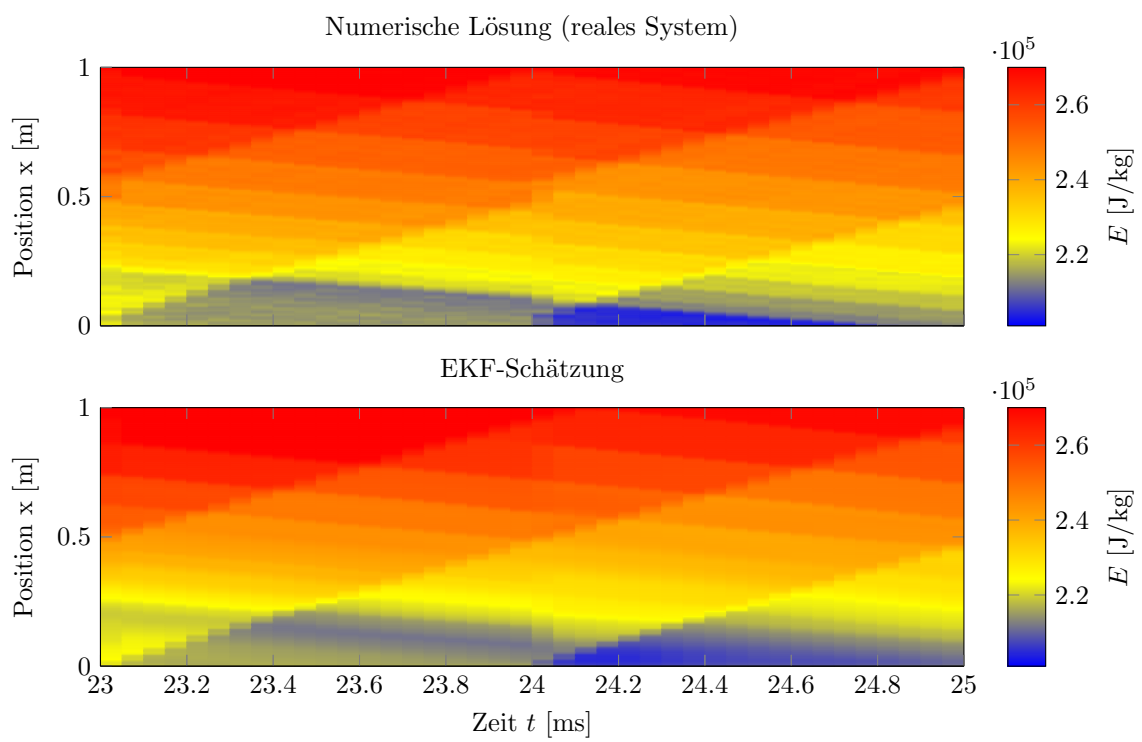


Abbildung 4.10: Intensitätsplot der numerischen Lösung und Schätzung der Energie. Für die Schätzung wurde das PINN-Modell ausgehend von den geschätzten Zuständen ausgewertet.

5 Modellprädiktive Regelung

Die modellprädiktive Regelung (MPC) ist ein leistungsfähiges Regelungsverfahren, welches für verschiedenste Aufgaben genutzt werden kann. Anwendungen der MPC reichen von Ausgangsfolgeaufgaben über Störunterdrückung bis zu allgemeiner Kostenoptimierung in der sogenannten ökonomischen MPC (Lee, 2011; Qin & Badgwell, 2003; S. Vazquez et al., 2014). Diese universelle Einsetzbarkeit der MPC sowie die umfangreichen, theoretischen Untersuchungen zu Stabilität und Konvergenzverhalten machen die MPC zu einem sehr vielseitigen Regelungsverfahren.

In diesem Abschnitt soll ein kurzer Überblick über die allgemeine (nichtlineare) MPC gegeben werden, bevor anschließend jeweils ein Beispiel für die Burgers-Gleichung und die Euler-Gleichungen auf Basis der in Abschnitt 3.4 beschriebenen Zustandsraummodellierung aufgebaut wird. Dieses Beispiel soll aufzeigen, wie gut der vorgestellte Ansatz in einem Regelungskontext eingesetzt werden kann, um ein komplexes System, dessen Dynamik mit partiellen Differentialgleichungen modelliert wurde, zu beeinflussen.

Ein ausführlicher Überblick über das Konzept der MPC ist z.B. in Morari et al., 1988 oder Rawlings, Meadows und Muske, 1994 zu finden. Die folgende Erklärung orientiert sich an den Ausführungen in Rawlings et al., 1994. Es wird sich auf den Fall mit einer skalaren Stellgröße und einem skalaren Ausgang beschränkt. Die grundlegende Idee der MPC ist die optimale Berechnung einer Stellfolge

$$\boldsymbol{\mu}_l^M = [\mu_l, \mu_{l+1}, \dots, \mu_{l+n_Y-1}]^T \quad (5.1)$$

im Zeitschritt l für die folgenden n_Y Zeitschritte.¹ Ausgehend von einem allgemeinen (nicht-linearen) diskreten Zustandsraummodell

$$\mathbf{x}_{l+1} = \mathbf{f}(\mathbf{x}_l, \mu_l) \quad (5.2)$$

$$y_l = \mathbf{z}(\mathbf{x}_l, \mu_l) \quad (5.3)$$

können damit sowohl die Zustände

$$\underline{\mathbf{x}}_l = [\mathbf{x}_{l+1}^T, \mathbf{x}_{l+2}^T, \dots, \mathbf{x}_{l+n_Y}^T]^T \quad (5.4)$$

als auch die Systemausgänge

$$\mathbf{y}_l = [y_{l+1}, y_{l+2}, \dots, y_{l+n_Y}]^T \quad (5.5)$$

prädiziert werden. Um bei der Lösung des im Folgenden erklärten Optimierungsproblems Rechenzeit zu sparen, wird die optimale Stellfolge nicht zwangsläufig für den gesamten Prädiktionshorizont n_Y berechnet. Stattdessen wird eine gekürzte Stellfolge

$$\boldsymbol{\mu}_l = [\mu_l, \mu_{l+1}, \dots, \mu_{l+n_U-1}]^T \quad (5.6)$$

¹Sollte die zur Verfügung stehende Zeit in einem Zeitschritt nicht ausreichen, um die Aktuation im Zeitschritt l zu bestimmen, muss die Stellfolge in Gl. (5.1) und alle folgenden Angaben um einen Zeitschritt verschoben werden.

5 Modellprädiktive Regelung

eingeführt, aus der die für die Prädiktion verwendete Stellfolge μ_l^M z.B. durch konstantes Fortsetzen bestimmt werden kann:

$$\mu_l^M = \left[\underbrace{\mu_l, \mu_{l+1}, \dots, \mu_{l+n_U-1}}_{\mu_l}, \underbrace{\mu_{l+n_U-1}, \dots, \mu_{l+n_Y-1}}_{n_Y - n_U \text{ Zeitschritte}} \right]^T. \quad (5.7)$$

Mit der Stellfolge und den prädizierten Werten für Ausgang y_l und Zustand \underline{x}_l kann zusammen mit weiteren Parametern (Referenz)

$$\mathbf{r}_l = [r_{l+1}, r_{l+2}, \dots, r_{l+n_Y}]^T \quad (5.8)$$

ein Kostenfunktional

$$J_l = J(\mathbf{r}_l, \underline{x}_l(\mu_l), y_l(\mu_l), \mu_l) \quad (5.9)$$

für den Zeitschritt l aufgestellt werden. Das Kostenfunktional bildet die Stellgröße und die prädizierten Größen auf einen skalaren Wert ab, der das Regelverhalten auf dem Prädiktionshorizont bewertet. Entsprechend kann die optimale Stellfolge für den Stellhorizont durch Lösen eines Optimierungsproblems

$$\mu_l = \arg \min_{\tilde{\mu}_l^M} J(\mathbf{r}_l, \underline{x}_l(\tilde{\mu}_l^M), y_l(\tilde{\mu}_l^M), \tilde{\mu}_l^M) \quad (5.10)$$

unter den Nebenbedingungen

$$\mathbf{x}_{l+i} = \mathbf{f}(\mathbf{x}_{l+i-1}, \mu_{l+i-1}) \quad (5.11)$$

$$y_{l+i} = \mathbf{z}(\mathbf{x}_{l+i}, \mu_{l+i}) \quad (5.12)$$

$$\forall i \in \{1, 2, \dots, n_Y\}$$

bestimmt werden. Zusätzliche Nebenbedingungen, beispielsweise um Stellgrößenbeschränkungen zu berücksichtigen, können ebenfalls in dem Optimierungsproblem bei Bedarf berücksichtigt werden. Von der optimalen Stellfolge μ_l wird anschließend der erste Eintrag μ_l ausgewählt und im Zeitschritt $l+1$ als Aktuation für das System verwendet. Im Folgenden Zeitschritt wird der gesamte Vorgang wiederholt.

Für MPC-Beispiele der Systeme auf Basis der Burgers-Gleichung bzw. der Euler-Gleichungen wird im Folgenden angenommen, dass die Zustände in jedem Zeitschritt l vollständig bekannt sind. Mit Blick auf die im vorangegangenen Kapitel gezeigten Ergebnisse der Zustandsschätzung für entsprechende Systeme ist dies eine valide Annahme.

Die Optimierung wird mit dem in der Matlab-Funktion *fmincon* implementierten Innere-Punkt-Verfahren gelöst. Dabei werden stets die Standardeinstellungen gemäß Mathworks, 2021 gewählt. Eine Erklärung des implementierten Innere-Punkt-Verfahrens kann in Byrd, Hribar und Nocedal, 1999 gefunden werden.

5.1 Regelung Burgers-Gleichung

Im Folgenden wird wieder angenommen, dass die numerische Lösung der Burgers-Gleichung das wahre Systemverhalten beschreibt. Um ein aktuiertes System auf Basis der Burgers-Gleichung aufzubauen, wird angenommen, dass die Randbedingung bei $x = -1$ genutzt werden kann um das System zu beeinflussen:

$$\mu(t) = h(-1, t). \quad (5.13)$$

Für den Aufbau des PINN-Modells wird die Aktuation auf den Bereich $\mu \in [-1, 1]$ beschränkt und für das Training in Schritten von 0.1 diskretisiert. Entsprechend den Beschreibungen in Abschnitt 3.4 wird folglich für jeden Aktuationsschritt in

$$\mathcal{M} = \{-1, -0.9, \dots, 0.9, 1\} \quad (5.14)$$

ein neuronales Netz trainiert. Die Netzstruktur und der Trainingsvorgang entsprechen für alle Aktuationsschritte dem Beispiel in Abschnitt 3.4.1. Dabei werden jeweils, wie in den Abschnitten 3.4.1 und 4.2 beschrieben, einhundert verschiedene Zustände als Eingang berücksichtigt. Die so antrainierten neuronalen Netze werden dann entsprechend Abschnitt 3.4 mit einer Interpolation verknüpft.

Um zu demonstrieren, wie gut das PINN-Modell im Kontext einer modellgestützten Regelung funktioniert, wird eine einfache Messgleichung

$$y_l = z(x_l, \mu_l) = x_3 \quad (5.15)$$

angenommen. Als Referenz wird ein zyklischer Verlauf

$$r(t) = A \cdot \sin(f_r \cdot 2\pi \cdot t) + b \quad (5.16)$$

über die Zeit definiert, dem der Ausgang des Systems y_l folgen soll. Dabei wird eine Amplitude $A = 0.4$, eine Frequenz $f_r = 0.02 \text{ Hz}$ und ein Offset $b = 0.4$ für die Referenz festgelegt. Die Güte des Folgeverhaltens wird mit der 2-Norm des Fehlers zwischen Referenz und Systemausgang über die n_Y Zeitschritte des Prädiktionshorizonts bewertet. Die Kostenfunktion gemäß Gl. (5.9) ergibt sich folglich zu

$$J_l = J(\mu_l) = \|\mathbf{r}_l - \mathbf{y}_l\|_2 \quad (5.17)$$

Zudem wird die Beschränkung der Aktuation $\mu \in [-1, 1]$ in der Optimierung berücksichtigt, da das PINN-Modell wie oben beschrieben nur für diesen Bereich trainiert wurde. Es wird eine Abtastung der Messung und der Aktuation mit einer Frequenz von $f_u = f_y = 2 \text{ Hz}$ angenommen. Der Prädiktionshorizont wurde zu $n_Y = 5$ und der Stellhorizont zu $n_U = 3$ gewählt. In den Untersuchungen zu diesem Beispiel konnte mit deutlich größeren Stell- bzw. Prädiktionshorizonten keine deutliche Verbesserung der Ergebnisse erzielt werden.

Abbildung 5.1 zeigt den Intensitätsplot in der Raum-Zeit-Ebene des mit der beschriebenen MPC aktuierten Systems. Dabei wurde erneut von einer sinusförmigen Anfangsbedingung, wie in den Abschnitten 3.4.1 und 4.2 beschrieben, ausgegangen. Der Intensitätsplot zeigt bereits nach kurzer Anfangszeit das gewünschte sinusförmige Verhalten. Dies gilt nicht nur an der Position, die dem eigentlichen Systemausgang entspricht, sondern über den gesamten Bereich $x \in [-1, 1]$. Das gute Folgeverhalten zeigt sich noch deutlicher, wenn ausschließlich der Systemausgang betrachtet wird. Abbildung 5.2 zeigt den zeitlichen Verlauf des Systemausgangs und der Referenz. Auch hier zeigt sich, dass die MPC in der Lage ist, dem Ausgang der Referenz bereits nach etwa $t = 2 \text{ s}$ zufriedenstellend zu folgen. Auffällig sind die stärkeren Abweichungen in den Spitzen und Senken der Referenz. Der Regelfehler in diesen Bereichen kann zum einen auf den zwar kleinen, aber vorhandenen Modellfehler zurückgeführt werden, zum anderen ist die Dynamik des Systems begrenzt, wodurch ein beliebig schnelles und genaues Folgen der Referenz erschwert wird.

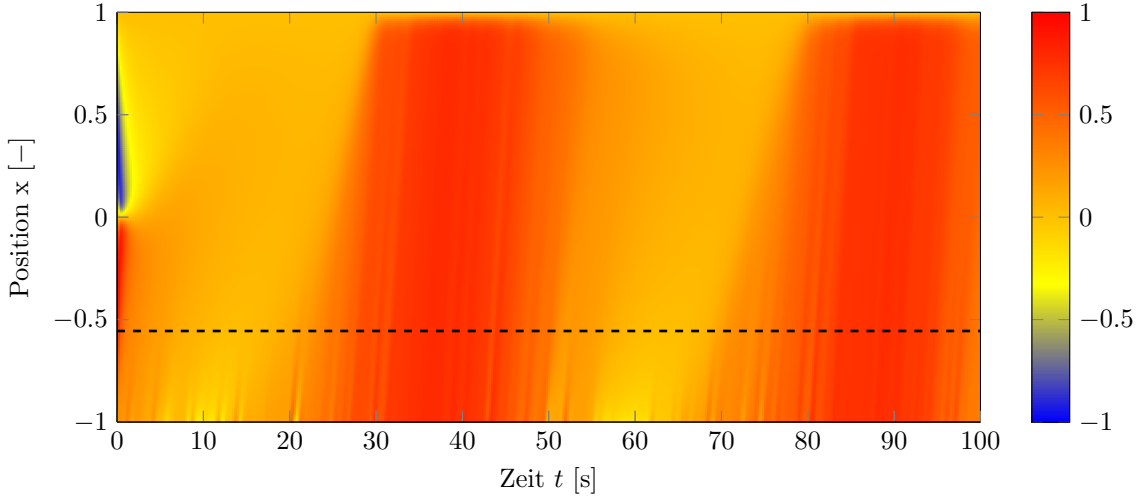


Abbildung 5.1: Intensitätsplot in der Raum-Zeit-Ebene eines durch eine MPC aktuierten Systems auf Basis der Burgers-Gleichung. Der Systemausgang der einer sinusförmigen Referenz folgen soll bezieht sich auf den durch die gestrichelte horizontale Linie gekennzeichneten Ort.

5.2 Regelung Euler-Gleichungen

Um die Leistungsfähigkeit der PINN-basierten Zustandsraummodellierung weiter zu verdeutlichen, wird in diesem Abschnitt auf ein MPC-Beispiel für ein System, welches durch die Euler-Gleichungen beschrieben wird, eingegangen. Als System wird die numerische 1-D-Simulation der Euler-Gleichungen angenommen, die bereits in Abschnitt 3.4.2 und Abschnitt 4.3 verwendet wurde. Das System entspricht der schematischen Darstellung in Abb. 4.4. Die Aktuation erfolgt über eine Anpassung des Drucks am Einlass des Rohres.

Ziel der MPC ist es, ähnlich zum Beispiel für die Burgers-Gleichung, mit der Geschwindigkeit $v_x(t, x = 0.3 \text{ m})$ an der Stelle $x = 0.3 \text{ m}$ im Rohr einer sinusförmigen Referenz

$$r(t) = b + A \cdot \sin(2\pi \cdot f_r \cdot t) \quad (5.18)$$

zu folgen. Dabei wird in diesem Beispiel eine Frequenz $f_r = 4 \text{ Hz}$, eine Amplitude $A = 25 \text{ m/s}$ und ein Offset $b = 175 \text{ m/s}$ für die Referenz angenommen. Die Messgleichung des Systems wird entsprechend so festgelegt, dass der Systemausgang ebenfalls der Geschwindigkeit an der Stelle $x = 0.3 \text{ m}$ entspricht:

$$y(t_l) = y_l = \xi \left(\Psi_{v_x}^{\mu_l}(0.3, 0, \mathbf{x}_l), \Psi_{v_x}^{\bar{\mu}_l}(0.3, 0, \mathbf{x}_l), \mu_l \right) \quad (5.19)$$

Als Kostenfunktion wird, wie im Beispiel der Burgers-Gleichung, die 2-Norm zwischen Systemausgang und Referenz festgelegt. Die Beschränkung der Änderungsrate des Eingangsdrucks wird über die lineare Ungleichung

$$\underbrace{\begin{bmatrix} 1 & -1 & \cdots & 0 & 0 \\ -1 & 1 & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & 1 & -1 \\ 0 & 0 & \cdots & -1 & 1 \\ 1 & 0 & \cdots & 0 & 0 \\ -1 & 0 & \cdots & 0 & 0 \end{bmatrix}}_{\Omega \in \mathbb{R}^{(n_Y \cdot 2) \times n_Y}} \cdot \underbrace{\begin{bmatrix} \mu_l \\ \vdots \\ \mu_{l+n_Y-1} \end{bmatrix}}_{\mu_l} \leq \underbrace{\begin{bmatrix} \Delta p \\ \Delta p \\ \vdots \\ \Delta p \\ \Delta p \\ \mu_{l-1} + \Delta p \\ \mu_{l-1} - \Delta p \end{bmatrix}}_{c_l} \quad (5.20)$$

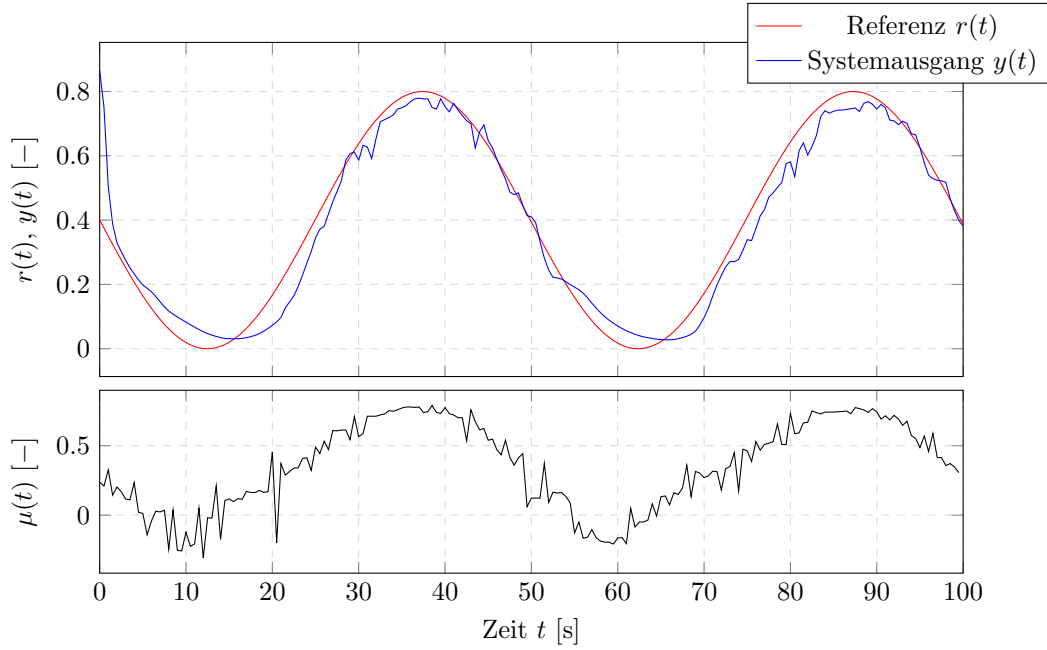


Abbildung 5.2: Folgeverhalten der MPC für ein System auf Basis der Burgers-Gleichung ausgehend von einer sinusförmigen Anfangsbedingung.
oben: Zeitlicher Verlauf des Systemausgangs und Referenz
unten: Zeitlicher Verlauf der Stellgröße

berücksichtigt. Hier bezeichnet $\Delta p = 0.1 \text{ bar/ms}$ die maximal mögliche Änderung des Drucks am Einlass des Rohres. Die letzten beiden Zeilen in Gl. (5.20) stellen sicher, dass die Änderung auch im ersten Schritt des aktuellen Horizonts beschränkt ist. Der Prädiktions- sowie Stellhorizont wurden in diesem Beispiel zu $n_Y = n_U = 5$ Zeitschritten gewählt. Untersuchungen mit größeren Prädiktions- und Stellhorizonten bis zu $n_Y = n_U = 50$ Zeitschritten haben keine deutliche Verbesserung ergeben und wurden aufgrund des hohen Rechenaufwands nicht weiter berücksichtigt. Die MPC in diesem Beispiel arbeitet auf einem Zeitgitter von $\Delta t = 1 \text{ ms}$. Dies entspricht dem maximalen Wert, der durch die trainierten Netze in einem Zeitschritt abgebildet werden kann.

In Abbildung 5.3 oben ist der Systemausgang aus der numerischen Simulation zusammen mit der Referenz dargestellt. Unten ist die durch die MPC berechnete Aktuation zu sehen. Als Anfangsbedingung wurde eine konstante Durchströmung mit 150 m/s und eine konstante Temperatur von $\vartheta = 300 \text{ K}$ im gesamten Rohr angenommen. Um eine Art Störung zu imitieren, wurde zudem für $t = 0$ eine sinusförmige Anfangsverteilung des Drucks

$$p(x, 0) = \sin(x) \quad (5.21)$$

über die Länge $x \in [0; 1]$ des Rohres in der Simulation berücksichtigt. Die Ergebnisse zeigen, dass es auch für die Euler-Gleichungen mit der MPC auf Basis eines PINN-Modells gelingt, einer Referenz zufriedenstellend zu folgen. In diesem konkreten Beispiel kann der Referenz bereits nach etwa 20 ms gut gefolgt werden. Von da an zeigt sich über die gesamte betrachtete Simulationzeitspanne eine sehr hohe Genauigkeit des Folgeverhaltens.

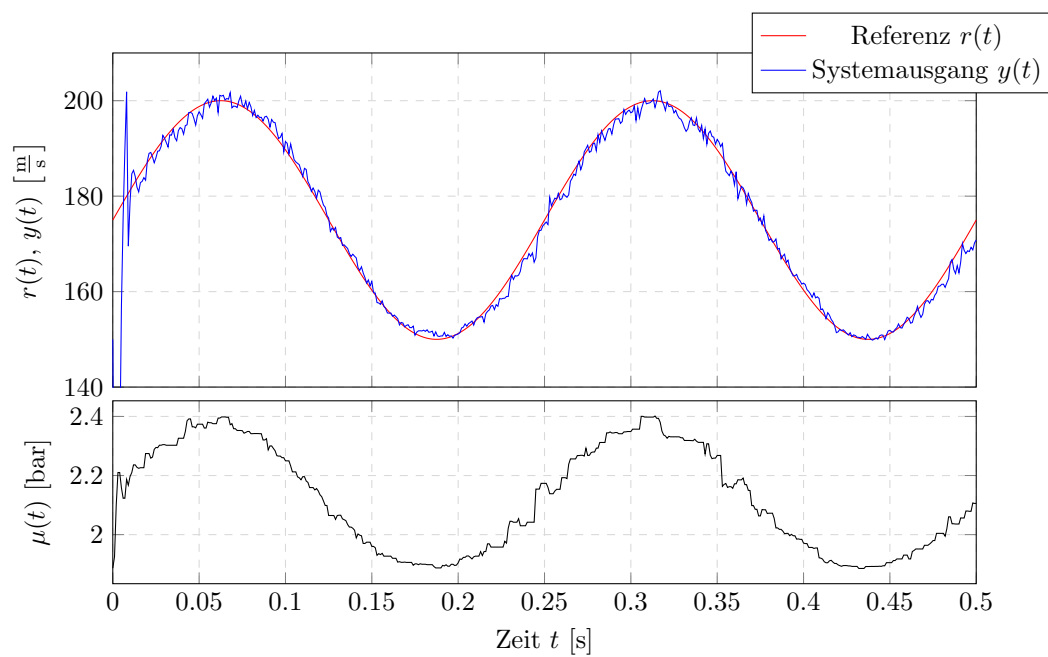


Abbildung 5.3: Folgeverhalten der MPC für ein System auf Basis der Euler-Gleichungen ausgehend von einer sinusförmigen Anfangsbedingung.
 oben: Zeitlicher Verlauf des Systemausgangs und Referenz
 unten: Zeitlicher Verlauf der Stellgröße

6 Iterativ lernende Regelung

In vielen technischen Prozessen wiederholen sich Abläufe mit einer gewissen Regelmäßigkeit. Wenn bestimmte Verläufe von Zuständen und Messgrößen sich zyklisch wiederholen oder Auswirkungen von zyklischen Störungen reduziert werden sollen, ist es sinnvoll, Information von vorangegangenen Zyklen in der Regelung zu nutzen, um z.B. das Folgeverhalten im Bezug auf die Referenz zu verbessern. Eine Möglichkeit hierfür bietet die sogenannte iterativ lernende Regelung (ILR).

Wie in der Einleitung beschrieben, existieren diverse technisch relevante Strömungsprozesse, die einen zyklischen Charakter zeigen. Daher ist die ILR ein besonders geeignetes Verfahren, um eine aktive Strömungsbeeinflussung mit einer geschlossenen Regelung aufzubauen. Kritisch sind die ebenfalls erwähnten mechanischen Beschränkungen in der Aktuation. Die beschränkte Bandbreite und der schaltende Charakter der Aktuatoren führt dazu, dass eine Reglersynthese im kontinuierlichen Bereich mitunter nicht mehr möglich ist oder zu unzufriedenstellenden Ergebnissen führt. Daher ist es sinnvoll, entsprechende Beschränkungen direkt in der Reglersynthese zu berücksichtigen.

In diesem Kapitel soll nach einer kurzen allgemeinen Einführung in das Konzept der ILR auf Beschränkungen in der Aktuation eingegangen werden und wie diese in einem ILR-Ansatz berücksichtigt werden können. Für eine einfachere Beschreibung wird in diesem Kapitel bis auf den letzten Abschnitt von Systemen mit einer Eingangs- und einer Ausgangsgröße ausgegangen.

6.1 Grundlagen der iterativ lernenden Regelung

Die grundlegende Idee einer ILR ist die Anpassung der Stellgröße $\mu(t)$ für einen ganzen Zyklus k basierend auf der Messung $y(t)$, die ebenfalls für einen gesamten Zyklus aufgenommen wurde. Ausgehend von einer Zyklusdauer T_p und einer Abtastzeit für die Aktuation mit der Frequenz f_u ergeben sich $n_u = T_p \cdot f_u$ Stellschritte

$$\mu_l = \mu(l \cdot \Delta t) \quad \text{mit} \quad \Delta t = 1/f_u \quad \text{für} \quad l \in \{0, \dots, n_u - 1\} \quad (6.1)$$

für einen Zyklus. Die Stellgrößen können dann für den Zyklus k in einem entsprechenden Vektor

$$\boldsymbol{\mu}_k = [\mu_{k,0}, \dots, \mu_{k,l}, \dots, \mu_{k,n_u-1}]^T \quad (6.2)$$

zusammengefasst werden. Analog ergeben sich für die Messung bei einer Messfrequenz von f_y in einem Zyklus $n_y = T_p \cdot f_y$ Messpunkte

$$y_l = y(l \cdot \Delta t) \quad \text{mit} \quad \Delta t = 1/f_y \quad \text{für} \quad l \in \{0, \dots, n_y - 1\}. \quad (6.3)$$

die ebenfalls in einem Vektor

$$\boldsymbol{y}_k = [y_{k,0}, \dots, y_{k,l}, \dots, y_{k,n_y-1}]^T \quad (6.4)$$

6 Iterativ lernende Regelung

zusammengefasst werden können, der aus allen Messung eines Zykluses k besteht.

Mit einer zyklusinvarianten Referenz $\mathbf{r} \in \mathbb{R}^{n_y}$ kann für den Zyklus k in Abhängigkeit der Messung der Regelfehler

$$\mathbf{e}_k = \mathbf{r} - \mathbf{y}_k \quad (6.5)$$

berechnet werden. Der Ansatz einer ILR ist ein Stellgesetz

$$\boldsymbol{\mu}_{k+1} = \mathbf{g}(\boldsymbol{\mu}_k, \boldsymbol{\mu}_{k-1}, \dots, \mathbf{e}_k, \mathbf{e}_{k-1}, \dots) \quad (6.6)$$

zu definieren, das die Stellfolge $\boldsymbol{\mu}_{k+1}$ für den kommenden Zyklus in Abhängigkeit der Stellfolgen und den Regelfehlern vorangegangener Zyklen definiert.

Für die Bestimmung des Stellgesetzes in Gl. (6.6) gibt es verschiedenste Ansätze. Dabei kann eine Reglersynthese sowohl im Zeitbereich als auch im Frequenzbereich erfolgen. Ein Überblick über die verschiedenen Methoden zur Bestimmung des Stellgesetzes Gl. (6.6) ist z.B. in Bristow, Tharayil und Alleyne, 2006 oder Ahn, Chen und Moore, 2007 zu finden. Im Rahmen dieser Arbeit wird sich auf den modellbasierten Ansatz der normoptimalen ILR beschränkt, die erstmals in Amann et al., 1998 vorgestellt wurde. Dafür ist ein Prozessmodell erforderlich, dessen Aufbau für den linearen Fall im Folgenden kurz beschrieben wird. Für den nichtlinearen Fall ist zunächst eine Linearisierung des Modells notwendig, um die Schritte zum Aufbau eines linearen Eingangs-/Ausgangsmodells für einen gesamten Zyklus durchführen zu können. Es ist auch die direkte Verwendung eines nichtlinearen Modells möglich, diese erfordert allerdings die Verwendung eines nichtlinearen Optimierungsverfahren. Dieser Fall wird hier nicht weiter betrachtet.

Für die folgende Beschreibung wird angenommen, dass die Abtastraten für Messung und Aktuation gleich sind, $f_u = f_y$. Es gilt also $n_u = n_y = n$. Für ein lineares System kann die Dynamik in einem diskreten Zustandsraummodell

$$\mathbf{x}_{l+1} = \mathbf{A}\mathbf{x}_l + \mathbf{B}\boldsymbol{\mu}_l \quad (6.7)$$

$$\mathbf{y}_l = \mathbf{C}\mathbf{x}_l + \mathbf{D}\boldsymbol{\mu}_l \quad (6.8)$$

aus Zustandsgleichung und Messgleichung mit $\mathbf{x} \in \mathbb{R}^{n_x}$, $\mathbf{A} \in \mathbb{R}^{n_x \times n_x}$, $\mathbf{B} \in \mathbb{R}^{n_x \times 1}$, $\mathbf{C} \in \mathbb{R}^{1 \times n_x}$ und $\mathbf{D} \in \mathbb{R}$ beschrieben werden. Die Systemdynamik ist über die Matrizen \mathbf{A} , \mathbf{B} , \mathbf{C} und \mathbf{D} definiert. Damit ergibt sich für die Zustände in den einzelnen Zeitschritten eines Zykluses

$$\mathbf{x}_1 = \mathbf{A}\mathbf{x}_0 + \mathbf{B}\boldsymbol{\mu}_0 \quad (6.9)$$

$$\mathbf{x}_2 = \mathbf{A}\mathbf{x}_1 + \mathbf{B}\boldsymbol{\mu}_1 = \mathbf{A}^2\mathbf{x}_0 + \mathbf{A}\mathbf{B}\boldsymbol{\mu}_0 + \mathbf{B}\boldsymbol{\mu}_1 \quad (6.10)$$

\vdots

$$\mathbf{x}_l = \mathbf{A}^l\mathbf{x}_0 + \mathbf{A}^{l-1}\mathbf{B}\boldsymbol{\mu}_0 + \dots + \mathbf{A}\mathbf{B}\boldsymbol{\mu}_{l-2} + \mathbf{B}\boldsymbol{\mu}_{l-1} \quad (6.11)$$

Daraus können die Systemausgänge

$$\mathbf{y}_0 = \mathbf{C}\mathbf{x}_0 + \mathbf{D}\boldsymbol{\mu}_0 \quad (6.12)$$

$$\mathbf{y}_1 = \mathbf{C}\mathbf{x}_1 + \mathbf{D}\boldsymbol{\mu}_1 = \mathbf{C}\mathbf{A}\mathbf{x}_0 + \mathbf{C}\mathbf{B}\boldsymbol{\mu}_0 + \mathbf{D}\boldsymbol{\mu}_1 \quad (6.13)$$

\vdots

$$\mathbf{y}_l = \mathbf{C}\mathbf{A}^l\mathbf{x}_0 + \mathbf{C}\mathbf{A}^{l-1}\mathbf{B}\boldsymbol{\mu}_0 + \dots + \mathbf{C}\mathbf{B}\boldsymbol{\mu}_{l-1} + \mathbf{D}\boldsymbol{\mu}_l \quad (6.14)$$

bestimmt und ein Eingangs-/Ausgangsmodell

$$\underbrace{\begin{bmatrix} y_{k,0} \\ y_{k,1} \\ \vdots \\ y_{k,n-1} \end{bmatrix}}_{\mathbf{y}_k} = \underbrace{\begin{bmatrix} \mathbf{D} & 0 & 0 & \cdots & 0 \\ \mathbf{CB} & \mathbf{D} & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \mathbf{CA}^{n-2}\mathbf{B} & \mathbf{CA}^{n-3}\mathbf{B} & \cdots & \mathbf{CB} & \mathbf{D} \end{bmatrix}}_{\mathbf{G}} \underbrace{\begin{bmatrix} \mu_{k,0} \\ \mu_{k,1} \\ \vdots \\ \mu_{k,n-1} \end{bmatrix}}_{\boldsymbol{\mu}_k} + \underbrace{\begin{bmatrix} \mathbf{C} \\ \mathbf{CA} \\ \vdots \\ \mathbf{CA}^{n-1} \end{bmatrix}}_{\mathbf{G}_0} \mathbf{x}_0 \quad (6.15)$$

aufgestellt werden, welches die Stellgrößen auf die Systemausgänge für einen gesamten Zyklus k abbildet. Wird die Anfangsbedingung \mathbf{x}_0 als ebenfalls zyklusinvariant angenommen, ist der zweite Summand auf der rechten Seite von Gl. (6.15) ein konstanter Versatz, der, wie unten gezeigt wird, durch die ILR vollständig kompensiert werden kann. Neben der Herleitung aus einem Zustandsraummodell, wie hier gezeigt, ist auch eine Modellierung mit experimentellen Daten z.B. über eine Anregung mit Impulsantworten möglich. Dieses Vorgehen wird weiter unten im Mehrgrößenfall Anwendungen finden. Auch dieser Weg führt auf ein Modell mit der in Gl. (6.15) gezeigten Struktur.

Annahme 1: Unabhängig von der Konstruktion bzw. Herleitung der Systemübertragungsmatrix \mathbf{G} wird im Folgenden davon ausgegangen, dass die Abbildung \mathbf{G} injektiv ist.

Annahme 2: Mit $\mathbf{D} \in \mathbb{R}$ und $\mathbf{D} \neq 0$ ist \mathbf{G} regulär und der Kern besteht nur aus dem Nullvektor. Für $\mathbf{D} = 0$ und $\mathbf{CB} = 0$ wird $\mathbf{y}_k = [y_{k,1}, y_{k,2}, \dots, y_{k,n-1}]^T$ und $\boldsymbol{\mu}_k = [\mu_{k,0}, \mu_{k,1}, \dots, \mu_{k,n-2}]^T$ gewählt, sodass \mathbf{G} auch dann regulär ist.

6.2 Normoptimale iterativ lernende Regelung für lineare Systeme

Die folgenden Ausführungen zur ganzzahligen, beschränkten ILR wurden in Arnold und King, 2021a veröffentlicht. Die Idee der normoptimalen ILR ist die optimale Anpassung der Stellfolge $\boldsymbol{\mu}_{k+1}$ bezüglich einer Kostenfunktion

$$J_{k+1} = J(\boldsymbol{\mu}_{k+1}) = \mathbf{e}_{k+1}^T \mathbf{W}_E \mathbf{e}_{k+1} + \boldsymbol{\mu}_{k+1}^T \mathbf{W}_U \boldsymbol{\mu}_{k+1} + \Delta \boldsymbol{\mu}_{k+1}^T \mathbf{W}_{\Delta U} \Delta \boldsymbol{\mu}_{k+1} \quad (6.16)$$

in jedem Zyklus k für den folgenden Zyklus $k+1$. Der erste Summand entspricht dem mit der Matrix $\mathbf{W}_E \in \mathbb{R}^{n \times n}$ gewichteten, quadratischen Regelfehler für den nächsten Zyklus $k+1$. Der zweite Summand beschreibt die durch die Matrix $\mathbf{W}_U \in \mathbb{R}^{n \times n}$ gewichtete, quadratische Stellfolge. Der Ausdruck kann folglich als Gewichtung der Stellenergie im Zyklus $k+1$ betrachtet werden. Der letzte Summand gewichtet mit der Matrix $\mathbf{W}_{\Delta U} \in \mathbb{R}^{n \times n}$ die Änderung der Stellfolge vom aktuellen Zyklus zum nächsten Zyklus $\Delta \boldsymbol{\mu}_{k+1} = \boldsymbol{\mu}_{k+1} - \boldsymbol{\mu}_k$. Diese Gewichtung ist sinnvoll, um zu große Änderungen der Stellfolge von Zyklus zu Zyklus zu verhindern, die zu einem schwingenden Verhalten führen können. Zudem erhöht sich damit die Robustheit gegenüber Störungen und Modellfehlern.

Annahme 3: Die Gewichtungsmatrizen \mathbf{W}_E , \mathbf{W}_U und $\mathbf{W}_{\Delta U}$ sind alle symmetrisch und positiv definit.

Da der Regelfehler \mathbf{e}_{k+1} im Zyklus k unbekannt ist, muss der Systemausgang \mathbf{y}_{k+1} mit einem Modell prädiziert werden. Wird eine zyklusinvariante, unbekannte Störung \mathbf{d} angenommen, gilt ausgehend von dem linear-affinen Prozessmodell aus Gl. (6.15)

$$\mathbf{y}_{k+1} = \mathbf{G} \boldsymbol{\mu}_{k+1} + \underbrace{\mathbf{G}_0 \mathbf{x}_0 + \mathbf{d}}_{\mathbf{b}} \quad (6.17)$$

6 Iterativ lernende Regelung

An dieser Stelle wird von einer zyklusinvarianten Anfangsbedingung ausgegangen. Im Offset \mathbf{b} können neben der Anfangsbedingung und Störungen auch Modellfehler berücksichtigt werden, solange diese ebenfalls zyklusinvariant sind. Mit der Modellannahme gilt für den Regelfehler

$$\mathbf{e}_{k+1} = \mathbf{r} - \mathbf{y}_{k+1} = \mathbf{r} - \mathbf{G}\boldsymbol{\mu}_{k+1} - \mathbf{b} \quad . \quad (6.18)$$

Da sowohl der Offset \mathbf{b} als auch die Referenz \mathbf{r} als zyklusinvariant angenommen wurden, können sie genutzt werden, um die Fehler zweier aufeinanderfolgender Zyklen in einen Zusammenhang

$$\mathbf{r} - \mathbf{b} = \mathbf{e}_{k+1} + \mathbf{G}\boldsymbol{\mu}_{k+1} = \mathbf{e}_k + \mathbf{G}\boldsymbol{\mu}_k \quad . \quad (6.19)$$

zu bringen. Daraus folgt

$$\mathbf{e}_{k+1} = \mathbf{e}_k + \mathbf{G}\boldsymbol{\mu}_k - \mathbf{G}\boldsymbol{\mu}_{k+1} \quad . \quad (6.20)$$

Anmerkung 1: Aufgrund von Annahme 1 und der zyklusinvarianten Anfangsbedingung \mathbf{x}_0 ergibt sich ein abweichender Regelfehler

$$\mathbf{e}_{k+1} = \mathbf{e}_k - \mathbf{G}(\boldsymbol{\mu}_{k+1} - \boldsymbol{\mu}_k) \quad (6.21)$$

im Vergleich zum aktuellen Regelfehler \mathbf{e}_k für alle $\boldsymbol{\mu}_{k+1} - \boldsymbol{\mu}_k \neq \mathbf{0}$ solange sich diese Differenz nicht im Nullraum von \mathbf{G} befindet. Aufgrund von Annahme 2 ist dies ausgeschlossen. Folglich können sich für unterschiedliche Stellfolge keine identischen Regelfehler ergeben.

Die Fehlerprädiktion aus Gl. (6.20) kann in Gl. (6.16) eingesetzt werden. Es ergibt sich der bezüglich der neuen Stellfolge $\boldsymbol{\mu}_{k+1}$ quadratische Ausdruck

$$J_{k+1} = \boldsymbol{\mu}_{k+1}^T \mathbf{Q} \boldsymbol{\mu}_{k+1} + \mathbf{q}^T \boldsymbol{\mu}_{k+1} + q \quad (6.22)$$

mit

$$\mathbf{Q} = \mathbf{G}^T \mathbf{W}_E \mathbf{G} + \mathbf{W}_U + \mathbf{W}_{\Delta U} \quad (6.23)$$

$$\mathbf{q} = -2 \left(\boldsymbol{\mu}_k^T (\mathbf{G}^T \mathbf{W}_E \mathbf{G} + \mathbf{W}_{\Delta U}) + \mathbf{e}_k^T \mathbf{W}_E \mathbf{G} \right) \quad (6.24)$$

$$q = \boldsymbol{\mu}_k^T \mathbf{W}_{\Delta U} \boldsymbol{\mu}_k + \mathbf{e}_k^T \mathbf{W}_E \mathbf{e}_k \quad . \quad (6.25)$$

Anmerkung 2: Die Gewichte \mathbf{W}_E , \mathbf{W}_U und $\mathbf{W}_{\Delta U}$ wurden als positiv definit angenommen. Zudem ist unter Annahme 1 auch der Ausdruck $\mathbf{G}^T \mathbf{W}_E \mathbf{G}$ positiv definit. Daraus folgt, dass auch der quadratische Parameter \mathbf{Q} als Summe dieser Ausdrücke positiv definit ist und das unbeschränkte Problem eine eindeutige Lösung hat. Eine ausführliche Erläuterung dazu kann in Nocedal und Wright, 2006 gefunden werden.

Die optimale Stellfolge $\boldsymbol{\mu}_{k+1}$ für den nächsten Zyklus ergibt sich im unbeschränkten Fall durch die analytische Berechnung des Minimums der quadratischen Kostenfunktion aus Gl. (6.22). Es folgt für das Stellgesetz

$$\boldsymbol{\mu}_{k+1} = \mathbf{Q} \boldsymbol{\mu}_k + \mathbf{L} \mathbf{e}_k \quad (6.26)$$

mit

$$\mathbf{Q} = \left(\mathbf{G}^T \mathbf{W}_E \mathbf{G} + \mathbf{W}_U + \mathbf{W}_{\Delta U} \right)^{-1} \left(\mathbf{G}^T \mathbf{W}_E \mathbf{G} + \mathbf{W}_{\Delta U} \right) \quad (6.27)$$

$$\mathbf{L} = \left(\mathbf{G}^T \mathbf{W}_E \mathbf{G} + \mathbf{W}_U + \mathbf{W}_{\Delta U} \right)^{-1} \mathbf{G}^T \mathbf{W}_E \quad . \quad (6.28)$$

Da die Filter \mathbf{Q} und \mathbf{L} für das Stellgesetz in Gl. (6.26) während der Reglerauslegung berechnet werden können und sich während der Anwendung nicht ändern, ist der Rechenaufwand einer normoptimalen ILR auf zwei Matrixmultiplikationen beschränkt. Damit ist dieser Ansatz besonders vorteilhaft für Anwendungen mit hohen Anforderungen an die Echtzeitfähigkeit, sprich im Fall hoher Abtastraten.

Für den Rest dieses Kapitel wird die Bestrafung der Stellenergie vernachlässigt. Entsprechend wird $\mathbf{W}_U = \mathbf{0}$ angenommen. Die Allgemeingültigkeit der folgenden Aussagen wird dadurch nicht beeinträchtigt. Die Kostenfunktion reduziert sich damit auf

$$J_{k+1} = J(\boldsymbol{\mu}_{k+1}) = \mathbf{e}_{k+1}^T \mathbf{W}_E \mathbf{e}_{k+1} + \Delta \boldsymbol{\mu}_{k+1}^T \mathbf{W}_{\Delta U} \Delta \boldsymbol{\mu}_{k+1} \quad . \quad (6.29)$$

Im unbeschränkten Fall ergibt sich $\mathbf{Q} = \mathbf{I}$ für das erste Filter im Stellgesetz aus Gl. (6.26), womit der Rechenaufwand für die Anpassung der Stellfolge in jedem Zyklus nochmal reduziert wird. In diesem Fall wird im konvergierten Zustand

$$\boldsymbol{\mu}_{k+1} - \boldsymbol{\mu}_k = \mathbf{0} = \mathbf{L} \mathbf{e}_k \quad (6.30)$$

ideales Folgeverhalten erreicht.

Anmerkung 3: Wie beschrieben verbessert das Gewicht $\mathbf{W}_{\Delta U}$ die Robustheit gegenüber Störungen und Modellfehlern und ermöglicht eine konservativere Anpassung der Stellfolge $\boldsymbol{\mu}_k$ über die Iterationen. Wird das Gewicht verkleinert, ergibt sich ein aggressiveres Lern- und damit schnelleres Konvergenzverhalten der ILR. Für den Fall $\mathbf{W}_{\Delta U} = \mathbf{W}_U = \mathbf{0}$ folgt daraus der Ansatz der Inversen-Modell-ILR (Bristow et al., 2006). Die Regelung springt dann in einem Schritt zur optimalen Lösung.

6.2.1 Beschränkungen

In einer realen Applikation ist die Stellfolge $\boldsymbol{\mu}_k$ üblicherweise beschränkt. Diese Beschränkungen können aus physikalischen Beschränkungen resultieren, wie z.B. durch die Verwendung von schaltenden Bauteilen. Aber auch Entwurfsentscheidungen, z.B. zur Limitierung der Stellenergie, können durch entsprechende Beschränkungen definiert werden. In diesem Kapitel wird angenommen, dass die Stellgröße in jedem Fall auf den Raum der ganzen Zahlen \mathbb{N}^{n_u} beschränkt ist. Wie in der Einleitung in Abschnitt 1.2 beschrieben, sind für eine schnelle Ak-tuation von Strömungssystemen oft schaltende Bauteile notwendig. Um diese Charakteristik möglichst gut in der Reglersynthese abzubilden, ist es sinnvoll, diese Beschränkung der Ak-tuation entsprechend zu berücksichtigen. Zusätzlich wird angenommen, dass möglicherweise weitere Beschränkungen existieren, die den Raum zulässiger Stellfolgen auf eine finite Menge

$$\mathbf{u}_k \in \mathcal{U} \subsetneq \mathbb{N}^{n_u} \quad \forall k \in \{0, 1, \dots\} \quad (6.31)$$

reduzieren. Um im Folgenden einfach zwischen unbeschränkten Stellfolgen und Stellfolgen, die gemäß den Beschränkung in Gl. (6.31) zulässig sind, zu unterscheiden, wird für den zweiten Fall die alternative Notation \mathbf{u}_k für die Stellfolge im Zyklus k eingeführt.

Wenn der Lösungsraum für die Stellfolge entsprechend Gl. (6.31) beschränkt ist, muss die optimale Stellfolge \mathbf{u}_{k+1} für den nächsten Zyklus $k + 1$ aus einem quadratischen Programm berechnet werden.

Quadratisches Programm (QP) 1: Die optimale Stellfolge \mathbf{u}_{k+1} einer beschränkten ILR für den folgenden Zyklus $k + 1$ ergibt sich aus

$$\mathbf{u}_{k+1} = \arg \min_{\tilde{\mathbf{u}}_{k+1}} \tilde{\mathbf{u}}_{k+1}^T \mathbf{Q} \tilde{\mathbf{u}}_{k+1} + \mathbf{q}^T \tilde{\mathbf{u}}_{k+1} \quad (6.32)$$

6 Iterativ lernende Regelung

unter der Beschränkung

$$\tilde{\mathbf{u}}_{k+1} \in \mathcal{U} \subsetneq \mathbb{N}^{n_u} \quad (6.33)$$

mit

$$\mathbf{Q} = \mathbf{G}^T \mathbf{W}_E \mathbf{G} + \mathbf{W}_{\Delta U} \quad (6.34)$$

$$\mathbf{q}^T = -2 \left(\mathbf{u}_k^T \left(\mathbf{G}^T \mathbf{W}_E \mathbf{G} + \mathbf{W}_{\Delta U} \right) + \mathbf{e}_k^T \mathbf{W}_E \mathbf{G} \right) \quad (6.35)$$

Im Gegensatz zu der unbeschränkten Formulierung zu Beginn dieses Abschnitts, kann das QP 1 nicht analytisch gelöst werden. Dadurch wird der Rechenaufwand für die Berechnung der Stellfolge in jedem Zyklus deutlich erhöht. Vor allem ist der Rechenaufwand aber nicht deterministisch. Es ist also vor der eigentlichen Anwendung nicht möglich zu garantieren, in welcher Zeitspanne das QP 1 optimal gelöst werden kann. Lediglich eine obere Abschätzung über die maximale Anzahl der zulässigen Lösungen $|\mathcal{U}|$ ist möglich. Dadurch ist eine echtzeitfähige Umsetzung analog zur reellwertigen Lösung in vielen Fällen nicht möglich.

Neben den Auswirkungen auf den Rechenaufwand beeinflusst die Beschränkung unter Umständen auch das Konvergenzverhalten der ILR deutlich. Dies kann dazu führen, dass die beschränkte Regelung zu einem auf den Regelfehler bezogenen, lokalen Minimum konvergiert und somit ein globales, optimales Folgeverhalten im auskonvergierten Zustand nicht gewährleistet ist. Dies ist auf die minimal mögliche Änderung

$$\|\mathbf{u}_{k+1} - \mathbf{u}_k\|_2 \geq 1 \quad \text{für} \quad \mathbf{u}_{k+1} \neq \mathbf{u}_k \quad (6.36)$$

zurückzuführen, die aus der ganzzahligen Beschränkung resultiert. Durch diese minimal möglichen Änderungen existiert auch eine untere Grenze

$$(\mathbf{u}_{k+1} - \mathbf{u}_k)^T \mathbf{W}_{\Delta U} (\mathbf{u}_{k+1} - \mathbf{u}_k) > w_{\Delta U, \min} \quad (6.37)$$

für den Bestrafungsterm der Änderung in der Kostenfunktion Gl. (6.16). Damit kann eine Situation entstehen, in der

$$J(\mathbf{u}_{k+1}) \geq J(\mathbf{u}_k) \quad \forall \mathbf{u}_{k+1} \in \mathcal{U} \quad (6.38)$$

gilt, obwohl möglicherweise eine Stellfolge $\mathbf{u}_{k+1} \in \mathcal{U}$ existiert, welche in einem Regelfehler mit

$$\mathbf{e}_{k+1}^T \mathbf{W}_E \mathbf{e}_{k+1} < \mathbf{e}_k^T \mathbf{W}_E \mathbf{e}_k \quad (6.39)$$

resultiert. Die ganzzahlige Beschränkung führt also dazu, dass möglicherweise die optimale Lösung bezüglich des Regelfehlers im auskonvergierten Zustand nicht erreicht werden kann. Im folgenden Abschnitt werden die wichtigsten Eigenschaften sowohl der reellwertigen ILR sowie der ganzzahligen ILR von einer formaleren Seite betrachtet.

6.2.2 Stabilität und Konvergenz

Für das zyklusdynamische Verhalten eines ILR-geregelten Systems können Stabilität und Konvergenzeigenschaften definiert werden. In einem ersten Schritt werden die Eigenschaften für den unbeschränkten, reellwertigen ILR-Ansatz rekapituliert. Im Anschluss werden entsprechende zweckdienliche Definitionen für den ganzzahligen Fall eingeführt, da entsprechende Annahmen aus dem unbeschränkten, reellwertigen Fall nicht direkt übertragbar sind. Diese zweckdienlichen Definitionen werden jeweils den Überlegungen für den unbeschränkten, reellwertigen Fall gegenübergestellt.

Definition 1: Eine reellwertige ILR ist asymptotisch stabil, wenn eine Grenze μ_{ub} existiert, sodass

$$|\mu_k| \leq \mu_{\text{ub}} \quad \forall k \in 0, 1, 2, \dots \quad (6.40)$$

und

$$\lim_{k \rightarrow \infty} \mu_k = \mu_{\infty} \quad (6.41)$$

gilt, vergleiche Bristow et al., 2006.

Behauptung 1: Die konvergierte Stellfolge μ_{∞} und der aus ihr resultierende Regelfehler \mathbf{e}_{∞} ist eindeutig für ein System.

Beweis: Die Eindeutigkeit der konvergierten Stellgröße μ_{∞} folgt direkt aus Anmerkung 2. Der konvergierte Regelfehler

$$\mathbf{e}_{\infty} = \mathbf{r} - \mathbf{b} - \mathbf{G}\mu_{\infty} \quad (6.42)$$

für $k \rightarrow \infty$ resultiert aus der eindeutigen optimalen Stellfolge μ_{∞} . \square

Definition 2: Eine reellwertige ILR konvergiert bezüglich einer gegebenen Norm $\|\cdot\|$ monoton genau dann, wenn

$$\|\mathbf{e}_{\infty} - \mathbf{e}_{k+1}\| \leq \gamma \|\mathbf{e}_{\infty} - \mathbf{e}_k\| \quad \forall k \quad \text{mit} \quad \gamma < 1, \quad (6.43)$$

wobei \mathbf{e}_{∞} wiederum der Regelfehler ist, der sich aus einer Aktuation mit der konvergierten Stellfolge μ_{∞} ergibt. Die Konvergenzrate $\gamma \in \mathbb{R}$ definiert dabei das Konvergenzverhalten, d.h. die Konvergenzgeschwindigkeit.

Behauptung 2: Monotone Konvergenz gemäß Definition 2 ergibt sich für eine unbeschränkte, reellwertige ILR, wenn

$$\gamma = \sigma_{\max}(\mathbf{G}(\mathbf{Q} - \mathbf{L}\mathbf{G})\mathbf{G}^{-1}) < 1 \quad (6.44)$$

erfüllt ist. Der Operator $\sigma_{\max}(\cdot)$ beschreibt hier den maximalen Singulärwert.

Beweis: Eine detaillierte Beschreibung des Beweises ist in Bristow et al., 2006 zu finden. Ausgangspunkt ist der bleibende Regelfehler \mathbf{e}_{∞} für den gilt

$$(\mathbf{I} + \mathbf{G}(\mathbf{I} - \mathbf{Q})^{-1}\mathbf{L})\mathbf{e}_{\infty} = \mathbf{r} - \mathbf{b}. \quad (6.45)$$

Für die Regelfehler in Zyklus k bzw. $k+1$ gilt

$$\mathbf{e}_k = \mathbf{r} - (\mathbf{G}\mu_k + \mathbf{b}) \quad (6.46)$$

$$\mathbf{e}_{k+1} = \mathbf{r} - (\mathbf{G}\mu_{k+1} + \mathbf{b}) \quad (6.47)$$

Durch Einsetzen des Stellgesetzes aus Gl. (6.26) in Gl. (6.47) ergibt sich ein Zusammenhang zwischen den Fehlern zweier aufeinander folgender Zyklen k und $k+1$:

$$\mathbf{r} - \mathbf{b} = \mathbf{e}_{k+1} + \mathbf{GQ}\mu_k + \mathbf{GL}\mathbf{e}_k \quad (6.48)$$

Gleichung (6.46) kann nach μ_k aufgelöst werden und in Gl. (6.48) eingesetzt werden. Es ergibt sich

$$\mathbf{r} - \mathbf{b} = (\mathbf{I} - \mathbf{GQ}\mathbf{G}^{-1})^{-1}(\mathbf{e}_{k+1} - \mathbf{G}(\mathbf{QG}^{-1} - \mathbf{L})\mathbf{e}_k) \quad (6.49)$$

Die linke Seite von Gl. (6.45) und die rechte Seite von Gl. (6.49) können gleichgesetzt werden und nach passender Umstellung der Gleichung ergibt sich

$$\mathbf{e}_\infty - \mathbf{e}_{k+1} = \mathbf{G}(\mathbf{Q} - \mathbf{L}\mathbf{G})\mathbf{G}^{-1}(\mathbf{e}_\infty - \mathbf{e}_k) \quad (6.50)$$

□

Anmerkung 4: Wie in Owens und Daley, 2008 beschrieben, ergibt sich für den hier betrachteten Fall mit $\mathbf{W}_U = \mathbf{0}$ die monotone Konvergenz einer normoptimalen ILR aus einer einfachen Verschachtelung (*interlacing*). Durch Einsetzen von $\boldsymbol{\mu}_{k+1} = \boldsymbol{\mu}_k$ in die Kostenfunktion Gl. (6.29) ergibt sich

$$\mathbf{e}_{k+1}^T \mathbf{W}_E \mathbf{e}_{k+1} \leq J_{k+1} \leq \mathbf{e}_k^T \mathbf{W}_E \mathbf{e}_k \quad (6.51)$$

Damit folgt $\mathbf{e}_{k+1}^T \mathbf{W}_E \mathbf{e}_{k+1} \leq \mathbf{e}_k^T \mathbf{W}_E \mathbf{e}_k$.

Da die Konvergenzbedingung in Gl. (6.44) auf den analytisch bestimmten Update-Filtern \mathbf{Q} und \mathbf{L} beruht, ist sie in dieser Form nicht auf den ganzzahligen Fall übertragbar. Daher werden im Folgenden einige zweckdienliche Aussagen für das zyklusdynamische Verhalten einer ganzzahligen ILR aufgestellt.

Lemma 1: Die Norm des gewichteten Regelfehlers $\|\mathbf{e}_k\|_{\mathbf{W}_E} = \|\mathbf{e}_k^T \mathbf{W}_E \mathbf{e}_k\|$ einer ganzzahligen ILR ist nicht ansteigend über die Zyklen.

Beweis: Da die Anmerkung 4 nicht auf der Annahme einer unbeschränkten Stellfolge beruht, gilt die Aussage auch für den Fall einer ganzzahligen Stellfolge. Es folgt, dass der gewichtete Regelfehler $\mathbf{e}_{k+1}^T \mathbf{W}_E \mathbf{e}_{k+1}$ über die Zyklen nicht ansteigt. □

Annahme 4: Wenn die optimale Lösung von QP 1 nicht zu einer Verbesserung der Kostenfunktion führt, im Speziellen, wenn der Fall $J(\mathbf{u}_{N+1}) = J(\mathbf{u}_N)$ für $\mathbf{u}_{N+1} \neq \mathbf{u}_N$ auftritt, wird angenommen, dass die Stellfolge \mathbf{u}_N für die folgenden Zyklen beibehalten wird und damit ein konvergierter Zustand erreicht ist. Der konvergierte Zustand der Regelung ist ebenfalls erreicht, wenn $J(\mathbf{u}_{N+1}) > J(\mathbf{u}_N)$ für alle zulässigen Lösungen $\mathbf{u}_{N+1} \in \mathcal{U}$ gilt.

Annahme 5: Es wird eine gleichmäßige Bestrafung der Änderung der Stellfolge angenommen. Es gilt folglich $\mathbf{W}_{\Delta U} = w_{\Delta U} \cdot \mathbf{I}$, womit sich der Bestrafungsterm aus der Kostenfunktion Gl. (6.29) zu

$$(\mathbf{u}_{k+1} - \mathbf{u}_k)^T \mathbf{W}_{\Delta U} (\mathbf{u}_{k+1} - \mathbf{u}_k) = w_{\Delta U} \cdot \|\mathbf{u}_{k+1} - \mathbf{u}_k\|_2^2 \quad (6.52)$$

reduziert. Hier ist \mathbf{I} die Einheitsmatrix.

Lemma 2: Eine ganzzahlige, normoptimale ILR mit einer Optimierung auf Basis einer Branch-and-Bound-Suche konvergiert in einen finalen Zustand.

Beweis: Zunächst muss ein periodisches Verhalten der ILR in der Zyklusdomäne ausgeschlossen werden. Das Auftreten einer periodischen Sequenz über die Zyklen würde einen konvergierten Zustand verhindern. Angenommen

- (a) es existiert tatsächlich ein $\mathbf{u}_{k+m} = \mathbf{u}_k$ für $m > 1$ und
- (b) gemäß der Verschachtelung aus Anmerkung 4 gilt für alle $i = 1, 2, \dots, m$ die Gleichheit der Fehler

$$\mathbf{e}_{k+i}^T \mathbf{W}_E \mathbf{e}_{k+i} = \mathbf{e}_k^T \mathbf{W}_E \mathbf{e}_k \quad . \quad (6.53)$$

Gemäß der Annahme 4 wählt der Optimierer nur eine neue Stellgröße $\mathbf{u}_{k+i} \neq \mathbf{u}_{k+i-1}$ aus, wenn die Kostenfunktion J_{k+i} abnimmt. Mit der Annahme 5 müsste sich eine Sequenz der Stellfolge

$$\begin{aligned} \|\mathbf{u}_k - \mathbf{u}_{k-1}\|_2^2 &> \|\mathbf{u}_{k+1} - \mathbf{u}_k\|_2^2 > \|\mathbf{u}_{k+2} - \mathbf{u}_{k+1}\|_2^2 \\ &> \dots > \|\mathbf{u}_{k+m} - \mathbf{u}_{k+m-1}\|_2^2 = \|\mathbf{u}_k - \mathbf{u}_{k+m-1}\|_2^2 \end{aligned} \quad (6.54)$$

ergeben. Diese Sequenz kann jedoch nie auftreten, da in der Optimierung als Folge auf die Stellfolge \mathbf{u}_{k-1} direkt das bestmögliche \mathbf{u}_k ausgewählt wird, ohne vorher suboptimale Lösungen auszuwählen. Folglich ist für diesen Fall ein periodisches Verhalten in der Zyklusdomäne ausgeschlossen.

Wird abweichend zu (b), zusätzlich zu (a), angenommen, dass es mindestens eine ILR-Iteration $k + j + 1$ gibt, in der für den Fehler

$$\mathbf{e}_{k+j+1}^T \mathbf{W}_E \mathbf{e}_{k+j+1} < \mathbf{e}_{k+j}^T \mathbf{W}_E \mathbf{e}_{k+j} \quad (6.55)$$

gilt, kann über die oben aufgeführte Argumentation ebenfalls ein periodisches Verhalten in der Zyklusdomäne ausgeschlossen werden.

Ein periodisches Verhalten ist nicht nur für die Stellgröße ausgeschlossen, sondern auch für den Regelfehler. Für $m > 1$ kann ein identischer Regelfehler $\mathbf{e}_{k+m} = \mathbf{e}_k$ aufgrund der injektiven Abbildung durch \mathbf{G} nur auftreten, wenn auch die Stellfolgen $\mathbf{u}_{k+m} = \mathbf{u}_k$ identisch sind. Dieser Fall wurde oben ausgeschlossen. Ein identischer Regelfehler zwischen zwei Iterationen kann nur für $m = 1$ auftreten. Dann ergibt sich $\mathbf{u}_{k+1} = \mathbf{u}_k = \mathbf{u}_N$ als die konvergierte Stellfolge mit entsprechendem Regelfehler.

Der verwendete Optimierungsalgorithmus stellt sicher, dass die positiv definite Kostenfunktion entweder abnimmt oder in einem konvergierten Zustand verbleibt. Damit ist instabiles Verhalten in der Zyklusdomäne ausgeschlossen. Mit einem vollständigen Durchlaufen des *Branch-and-Bound*-Suchbaums kann folglich die optimale Lösung \mathbf{u}_k in jeder Iteration bestimmt werden und Konvergenz zu einem finalen Zustand ist sichergestellt. \square

Im Gegensatz zu Behauptung 1 für den reellwertigen Fall ist die Eindeutigkeit des konvergierten Zustands im ganzzahligen Fall nicht gegeben.

Lemma 3: Der konvergierte Regelfehler \mathbf{e}_N , der sich aus der ganzzahligen Aktuation \mathbf{u}_N ergibt, liegt in einem Bereich, der durch

$$(\mathbf{u}_{N+1} - \mathbf{u}_N)^T \mathbf{Q} (\mathbf{u}_{N+1} - \mathbf{u}_N) - 2\mathbf{e}_N^T \mathbf{W}_E \mathbf{G} (\mathbf{u}_{N+1} - \mathbf{u}_N) \geq 0 \quad (6.56)$$

für alle $\mathbf{u}_{N+1} \neq \mathbf{u}_N$ und $\mathbf{u}_{N+1}, \mathbf{u}_N \in \mathcal{U} \subset \mathbb{N}^{n_u}$ definiert ist.

Beweis: Aufgrund von Annahme 4 ist ein konvergierter Zustand erreicht, wenn

$$\mathbf{u}_{N+1}^T \mathbf{Q} \mathbf{u}_{N+1} + \mathbf{q}^T \mathbf{u}_{N+1} \geq \mathbf{u}_N^T \mathbf{Q} \mathbf{u}_N + \mathbf{q}^T \mathbf{u}_N \quad (6.57)$$

für alle $\mathbf{u}_{N+1} \neq \mathbf{u}_N$ aus der Menge der zulässigen Lösungen $\mathbf{u}_{N+1}, \mathbf{u}_N \in \mathcal{U}$ gilt. Durch Umformen und unter Beachtung der speziellen Struktur von Gl. (6.34) bzw. Gl. (6.35) ergibt sich eine Menge an Ungleichungen

$$\Delta \mathbf{u}_{N+1}^T \mathbf{Q} \Delta \mathbf{u}_{N+1} - 2\mathbf{e}_N^T \mathbf{W}_E \mathbf{G} \Delta \mathbf{u}_{N+1} \geq 0 \quad (6.58)$$

für alle zulässigen Änderungen $\Delta \mathbf{u}_{N+1} = \mathbf{u}_{N+1} - \mathbf{u}_N$, womit Gl. (6.56) bewiesen ist. \square

Anmerkung 5: Da der quadratische Parameter \mathbf{Q} positiv definit ist, ist ein verschwindender Regelfehler $\mathbf{e}_N = \mathbf{0}$ in jedem Fall eine Lösung von Gl. (6.56). Wenn für $\mathbf{u}_{N+1} \neq \mathbf{u}_N \in \mathcal{U}$ ein anderer Regelfehler $\mathbf{e}_N \neq \mathbf{0}$ möglich ist, existiert unter der Annahme eines Lipschitz-kontinuierlichen Systems ein initialer Regelfehler \mathbf{e}_0 , mit dem sich eine Sequenz

$$\mathbf{e}_N \leftarrow \mathbf{e}_{N-1} \leftarrow \cdots \leftarrow \mathbf{e}_0 \quad (6.59)$$

ergibt, die im Regelfehler \mathbf{e}_N endet. Die Sequenz bzw. der Regelfehler \mathbf{e}_0 ist abhängig vom Gewicht $\mathbf{W}_{\Delta U}$. Folglich ist der konvergierte Regelfehler nicht eindeutig.

Anmerkung 6: Da der quadratische Parameter $\mathbf{Q} = \mathbf{G}^T \mathbf{W}_E \mathbf{G} + \mathbf{W}_{\Delta U}$ und $\mathbf{W}_{\Delta U}$ selbst positiv definit sind, kann über eine Vergrößerung von $\mathbf{W}_{\Delta U}$ der Lösungsraum aus Gl. (6.56) beliebig vergrößert werden. Dies bedeutet, dass für eine größere Anzahl an Stellfolgen \mathbf{u}_N keine weiteren Anpassungen möglich sind.

Anmerkung 7: Aufgrund von Lemma 2 und der in der Optimierung verwendeten *Branch-and-Bound*-Suche wird die Menge aus Gl. (6.56), in der sich der konvergierte Regelfehler befindet, in einer endlichen Anzahl von N Zyklen erreicht. Da $\mathbf{u}_k \in \mathcal{U}$, gilt Definition 1 auch für den ganzzahligen Fall.

Für die reellwertige ILR ergibt sich mit dem Verweis auf Anmerkung 3 eine vom Gewicht $\mathbf{W}_{\Delta U}$ unabhängige Stellfolge $\boldsymbol{\mu}_\infty$. Im Unterschied dazu ist im Fall einer ganzzahlig beschränkten Stellfolge das Gewicht $\mathbf{W}_{\Delta U}$ der Grund, dass keine eindeutige, konvergierte Stellfolge $\mathbf{u}_N = \mathbf{u}_\infty$ erreicht wird. Daher ist der Vergleich zwischen den beiden Ansätzen schwierig, wenn ein $\mathbf{W}_{\Delta U} \neq \mathbf{0}$ gewählt wird. Die selbe eindeutige Stellfolge $\boldsymbol{\mu}_\infty$ resultiert auch aus einem auf die Kostenfunktion

$$J_\infty = \mathbf{e}_\infty^T \mathbf{W}_E \mathbf{e}_\infty \quad (6.60)$$

bezogenen, optimalen Regelfehler \mathbf{e}_∞ , der nach einer Iteration erreicht wird. Für den ganzzahligen Fall kann eine analoge Lösung bestimmt werden, die unten als Referenz für die erreichbare Regelgüte genutzt wird.

Quadratisches Programm (QP) 2: Die analog zu Gl. (6.60) optimale, eindeutige, ganzzahlige Stellfolge \mathbf{u}_∞ , die der optimalen Steuerung für das System entspricht, ergibt sich aus

$$\mathbf{u}_\infty = \arg \min_{\tilde{\mathbf{u}}_\infty} \tilde{\mathbf{u}}_\infty^T \mathbf{Q} \tilde{\mathbf{u}}_\infty + \mathbf{q}^T \tilde{\mathbf{u}}_\infty \quad (6.61)$$

unter der Beschränkung

$$\tilde{\mathbf{u}}_\infty \in \mathcal{U} \subsetneq \mathbb{N}^{n_u} \quad (6.62)$$

mit

$$\mathbf{Q} = \mathbf{G}^T \mathbf{W}_E \mathbf{G} \quad (6.63)$$

$$\mathbf{q}^T = -2 \left(\mathbf{u}_0^T \mathbf{G}^T \mathbf{W}_E \mathbf{G} + \mathbf{e}_0^T \mathbf{W}_E \mathbf{G} \right) \quad (6.64)$$

Die Stellfolge \mathbf{u}_0 entspricht einer beliebigen initialen Stellfolge mit einem entsprechenden Regelfehler \mathbf{e}_0 .

6.3 Synthese einer ganzzahligen iterativ lernenden Regelung

In diesem Abschnitt werden auf Basis der vorangegangenen formalen Untersuchungen einer ganzzahligen ILR die Möglichkeiten einer Reglersynthese diskutiert. Aus den beschriebenen

Eigenschaften lassen sich die folgenden, wünschenswerten Kriterien für eine ganzzahlige ILR ableiten. Eine ganzzahlige ILR sollte

- eine möglichst kleine obere Schranke für die Kostenfunktion sicherstellen,
- die Norm des Regelfehlers \mathbf{e}_k über die Zyklen monoton verringern und
- deterministischen Rechenaufwand für jeden Zyklus garantieren.

In einem ersten Schritt werden zwei vorhandene Ad-hoc-Ansätze kurz vorgestellt und bezüglich der gewünschten Kriterien diskutiert. Beide Ansätze wurden als praktikable Verfahren für die Regelung einer SEC entwickelt und an einem nichtreaktiven Versuchsaufbau getestet (Schäpel et al., 2018). Im Anschluss wird dann ein neues Syntheseverfahren vorgestellt, welches ausgehend von den genannten Kriterien für eine ganzzahlige ILR entwickelt wurde.

Zum besseren Verständnis werden die Ausführungen zu den Syntheseverfahren in den folgenden Abschnitten von einem konsistenten Beispiel begleitet. Damit soll der Vergleich zwischen den einzelnen Verfahren erleichtert werden. Für eine einfache Visualisierung wird eine zweidimensionale Stellfolge $\mathbf{u}_k \in \mathbb{R}^2$ angenommen. Zudem gilt $\mathbf{r} - \mathbf{b} = \mathbf{G} \mathbf{u}_k + \mathbf{e}_k = \mathbf{0}$. Damit ergibt sich die reduzierte Kostenfunktion

$$J_{k+1} = \mathbf{u}_{k+1}^T \mathbf{G}^T \mathbf{W}_E \mathbf{G} \mathbf{u}_{k+1} + (\mathbf{u}_{k+1} - \mathbf{u}_k)^T \mathbf{W}_{\Delta U} (\mathbf{u}_{k+1} - \mathbf{u}_k). \quad (6.65)$$

In diesem einfachen Beispiel werden Matrizen

$$\mathbf{G} = \begin{bmatrix} 0.7186 & 0 \\ 1.5809 & 0.7186 \end{bmatrix}, \quad \mathbf{W}_E = \begin{bmatrix} 1.1232 & 0 \\ 0 & 0.9683 \end{bmatrix}, \quad \mathbf{W}_{\Delta U} = 2 \mathbf{I} \quad (6.66)$$

angenommen, mit denen sich der quadratische Parameter des Optimierungsproblems QP 1 zu

$$\mathbf{Q} = \begin{bmatrix} 3 & 1.1 \\ 1.1 & 0.5 \end{bmatrix} + \mathbf{W}_{\Delta U} \quad (6.67)$$

ergibt. Die Beschränkung auf den zweidimensionalen Fall erlaubt die einzelnen Schritte der ILR im Folgenden in einfachen Konturplots darzustellen.

6.3.1 Ad-hoc-Ansätze

Die beiden in diesem Abschnitt diskutierten Ansätze wurden in Schäpel et al., 2018 vorgestellt. Beide Ansätze bieten Lösungen, um die durch Gl. (6.56) definierte Menge möglichst klein zu halten und so das Konvergenzverhalten der ganzzahligen ILR zu verbessern. Allerdings wurden die Methoden nicht mit dem Fokus auf den oben vorgestellten Kriterien für das Verhalten einer ganzzahligen ILR entwickelt. Um den Vergleich zu der weiter unten vorgestellten Optimale-Menge-Synthese (OMS) zu erleichtern, wird an dieser Stelle trotzdem Bezug auf die einzelnen Kriterien genommen.

Adaptives Gewicht $\mathbf{W}_{\Delta U}$

Wie in Abschnitt 6.2.1 beschrieben, besteht in Abhängigkeit des gewählten Gewichts $\mathbf{W}_{\Delta U}$ die Möglichkeit, dass ein Zyklus erreicht wird, in dem die Bestrafung der Änderung der Stellgröße die daraus zu erwartende Reduzierung des Regelfehlers übersteigt. Es wird davon

ausgegangen, dass initial ein Gewicht $\mathbf{W}_{\Delta U} \neq \mathbf{0}$ gewählt wird, um eine zu starke Anpassung der Stellfolge und damit unter Umständen verbundene Schwingungen im System zu vermeiden. Wie in Anmerkung 5 beschrieben, variiert je nachdem wie groß das Gewicht $\mathbf{W}_{\Delta U}$ gewählt wird, der Zyklus, in dem die konvergierte Stellfolge mit dem zugehörigen Regelfehler erreicht wird. Die konvergierte Stellfolge ist jedoch nicht zwangsläufig optimal bezüglich des Regelfehlers. Eine intuitive Lösung ist eine Anpassung des Gewichts $\mathbf{W}_{\Delta U}$, um eine weitere Verbesserung des Regelfehlers zu ermöglichen. Wird ein Zyklus erreicht, in dem eine weitere Anpassung der Stellgröße \mathbf{u}_k nicht mehr möglich ist, kann mit einem entsprechenden Adaptionsfaktor $0 < p_a < 1$ das Gewicht angepasst werden:

$$\mathbf{W}_{\Delta U}^{k+1} = \mathbf{W}_{\Delta U}^k \cdot p_a \quad . \quad (6.68)$$

Um die Anpassung im Zyklus k zu veranschaulichen, sind in Abb. 6.1 die Kostenfunktion für verschiedene Werte des Adaptionsfaktor p_a ausgehend von einer aktuellen Stellgröße $\mathbf{u}_k = [-1, 3]^T$ visualisiert. Es ist zu erkennen, dass ohne eine Anpassung des Gewichts $\mathbf{W}_{\Delta U}$

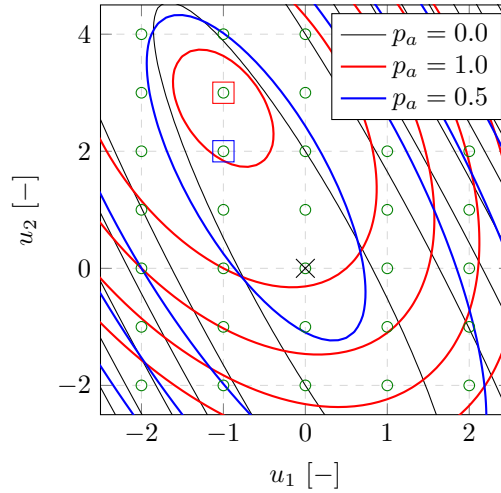


Abbildung 6.1: Konturplot für eine ganzzahlige ILR mit adaptivem Gewicht $\mathbf{W}_{\Delta U}$. Alle zulässigen Lösungen sind mit grünen Kreisen markiert. Der schwarze Konturplot visualisiert die Höhenlinien der Kostenfunktion für eine Einstellung ohne Bestrafung der Änderung von \mathbf{u}_k entsprechend QP 2. Dies führt zu einer optimalen Steuerung mit der optimalen Stellfolge $\mathbf{u}_\infty = \mathbf{0}$, die mit einem \times markiert ist. Der rote Konturplot zeigt die Kostenfunktion ohne Anpassung des Gewichts $\mathbf{W}_{\Delta U}$ und der blaue Konturplot zeigt die Kostenfunktion mit einem reduzierten Gewicht $\mathbf{W}_{\Delta U}^{k+1} = \mathbf{W}_{\Delta U}^k \cdot 0.5$. Die jeweiligen optimalen Stellgrößen sind mit einem Viereck markiert.

die optimale Stellgröße im Zeitschritt $k+1$ wiederum $\mathbf{u}_{k+1} = \mathbf{u}_k = [-1, 3]^T$ ist. Dies bedeutet, dass in diesem Punkt die konvergierte Stellfolge $\mathbf{u}_k = \mathbf{u}_N$ erreicht wäre. Eine weitere Verbesserung der Stellfolge ist ohne Veränderung der Gewichte nicht möglich. Der Vergleich mit der Lösung der optimalen Steuerung entsprechend QP 2 zeigt jedoch, dass die ILR-Lösung bezüglich der Kosten des Regelfehlers nur suboptimal ist.

Ein angepasstes Gewicht mit dem Adaptionsfaktor $p = 0.5$ ermöglicht eine aggressivere Einstellung der Stellgröße. Wie im Konturplot zu erkennen ist, verschiebt sich dadurch das Optimum des entsprechend angepassten QPs. So ergibt sich eine neue optimale Stellgröße $\mathbf{u}_{k+1} = [-1, 2]^T$, die in reduzierten Kosten des Regelfehlers resultiert. Letzteres erkennt man am schwarzen Konturplot, der auf den Kosten $J_k = \mathbf{e}_k^T \mathbf{W}_E \mathbf{e}_k$ aufbaut.

Die Anpassung des Gewichts kann über beliebig viele Iterationen wiederholt werden und dadurch das Gewicht $\mathbf{W}_{\Delta U}$ beliebig reduziert werden. Allerdings kann ein sehr kleines Gewicht $\mathbf{W}_{\Delta U}$ zu Problemen führen, wenn unbekannte Störungen oder plötzliche Wechsel in

der Referenz auftreten. Da ein größeres Gewicht $\mathbf{W}_{\Delta U}$ im Normalfall eine zu starke Anpassung der Stellgröße und damit ein mögliches Überschwingen des Systems verhindert, kann ein reduziertes Gewicht genau dieses unerwünschte Verhalten begünstigen. Daher sollte in der Anpassung auch eine Vergrößerung über einen Adaptionsfaktor $p_a > 1$ vorgesehen werden, der im gewünschten konvergierten Zustand die Reduzierung des Gewichts wieder rückgängig macht.

Bezogen auf die Kriterien für eine ganzzahlige ILR ist für diesen Ansatz Folgendes festzuhalten:

- **Fehlerschranke:** Da das Gewicht $\mathbf{W}_{\Delta U}$ über die Iterationen immer weiter reduziert werden kann, ergibt sich der Grenzwert $\mathbf{W}_{\Delta U} \rightarrow \mathbf{0}$. In diesem Fall wird das Optimalsteuerungsproblem aus QP 2 gelöst, welches die optimale Stellfolge bezüglich der Kosten des Regelfehlers liefert. Damit entspricht die Fehlerschranke genau den Kosten mit diesem Regelfehler.
- **Monotone Konvergenz:** Die in Abschnitt 6.2.2 beschriebene monotone Konvergenz einer ganzzahligen ILR ist nicht beeinträchtigt durch eine Verkleinerung oder Vergrößerung des Gewichts $\mathbf{W}_{\Delta U}$.
- **Deterministischer Rechenaufwand:** Der Rechenaufwand ist durch die Verwendung der *Branch-and-Bound*-Suche nur nach oben abschätzbar. Im schlechtesten Fall muss der gesamte Suchbaum exploriert werden, was bei hochdimensionalen Stellfolgen im Rahmen einer Iteration unter Echtzeitbedingungen meistens unmöglich ist. Der tatsächliche Rechenaufwand der *Branch-and-Bound*-Suche in einer Iteration ist nichtdeterministisch und variiert stark in Abhängigkeit des Gewichts $\mathbf{W}_{\Delta U}$.

Parallele reellwertige ILR

Da wie oben diskutiert die Anpassung des Gewichts $\mathbf{W}_{\Delta U}$ bei unbekannten Störungen die Gefahr von unvorhersehbaren Oszillationen birgt, wurde in Schäpel et al., 2018 ein weiterer Ansatz zur Verbesserung des Konvergenzverhaltens vorgeschlagen. Wie bereits oben diskutiert, kann es in einer ganzzahligen ILR dazu kommen, dass die minimal zulässige Änderungen der Stellfolge $\Delta \mathbf{u}_{k+1} \neq \mathbf{0}$ in einer größeren Bestrafung resultiert, als der durch die Anpassung verringerte Kostenanteil durch den Regelfehler. Der in Schäpel et al., 2018 vorgestellte Ansatz schlägt als direkte Lösung für dieses Problem vor, eine parallele, reellwertige ILR zu implementieren, deren virtuelle Stellgröße $\boldsymbol{\mu}_k$ in der Berechnung von

$$\Delta \mathbf{u}_{k+1} = \mathbf{u}_{k+1} - \boldsymbol{\mu}_k \quad (6.69)$$

berücksichtigt werden kann. Damit ändert sich das zu lösende QP.

Quadratisches Programm (QP) 3: Die Stellgröße einer ganzzahligen ILR mit der Berücksichtigung einer virtuellen, reellwertigen Stellfolge $\boldsymbol{\mu}_k \in \mathbb{R}^{n_u}$ ergibt sich im Zeitschritt k aus der Lösung von

$$\mathbf{u}_{k+1} = \arg \min_{\tilde{\mathbf{u}}_{k+1}} \tilde{\mathbf{u}}_{k+1}^T \mathbf{Q} \tilde{\mathbf{u}}_{k+1} + \mathbf{q}^T \tilde{\mathbf{u}}_{k+1} \quad (6.70)$$

unter der Beschränkung

$$\tilde{\mathbf{u}}_{k+1} \in \mathcal{U} \subset \mathbb{N}^{n_u} \quad (6.71)$$

mit

$$\mathbf{Q} = \mathbf{G}^T \mathbf{W}_E \mathbf{G} + \mathbf{W}_{\Delta U} \quad (6.72)$$

$$\mathbf{q}^T = -2 \left(\mathbf{u}_k^T \mathbf{G}^T \mathbf{W}_E \mathbf{G} + \boldsymbol{\mu}_k^T \mathbf{W}_{\Delta U} + \mathbf{e}_k^T \mathbf{W}_E \mathbf{G} \right) \quad (6.73)$$

Die Berechnung der reellwertigen Stellgröße $\boldsymbol{\mu}_k$ erfolgt nach dem allgemeinen Ansatz der normoptimalen ILR, der in Abschnitt 6.2 beschrieben ist. Dabei werden die Gewichtungen identisch zu dem ganzzahligen Aufbau gewählt. Da die minimale Änderung der reellwertigen Stellgröße infinitesimal klein ist, kann die Anpassung weitergeführt werden, wenn die ganzzahlige Stellfolge nicht mehr geändert werden kann.

Abbildung 6.2 illustriert wieder die Situation ausgehend von der Stellfolge $\mathbf{u}_k = [-1, 3]^T$. Auch wenn die Anpassung der ganzzahligen Stellfolge in einem Zeitschritt $k + 1$ nicht fortge-

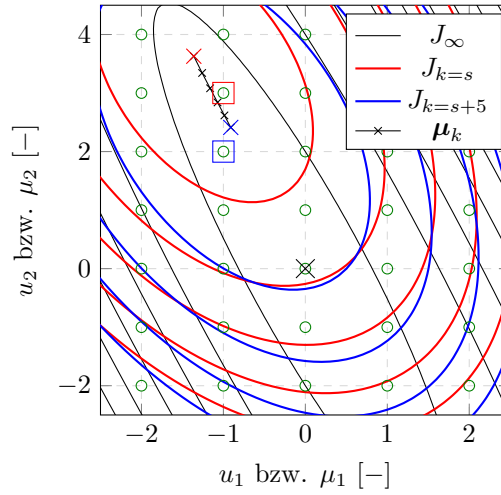


Abbildung 6.2: Konturplot für eine ganzzahlige ILR mit einer parallelen, reellwertigen ILR. Zulässige Lösungen sind mit grünen Kreisen markiert. Der rote Konturplot zeigt die Kostenfunktion in einem Zeitschritt $k = s$. Der blaue Konturplot zeigt die Kostenfunktion im Zyklus $k = s + 5$. Die jeweiligen ganzzahligen Optima sind mit Vierecken markiert. Die virtuelle Progression der reellwertigen Stellgröße in den Zyklen $s, s + 1, \dots, s + 5$ ist durch Kreuze dargestellt.

setzt wird, kann die Anpassung der virtuellen Stellfolge in den folgenden Zyklen fortgesetzt werden. Durch die Änderung der reellwertigen Stellfolge $\boldsymbol{\mu}_k$ ändert sich der lineare Parameter \mathbf{q} in QP 3. Dadurch verschiebt sich das Optimum über die Zyklen solange, bis eine weitere Anpassung möglich ist. In dem gezeigten Beispiel verschiebt sich in diesen Iterationen die Kostenfunktion, sodass nach fünf Zyklen eine weitere Anpassung der ganzzahligen Stellfolge möglich ist. Abbildung 6.3 zeigt den Verlauf der virtuellen, reellwertigen und ganzzahligen Stellgröße über einen größeren Zeitraum. Es ist zu erkennen, dass die reellwertige Stellfolge aus deutlich mehr Schritten besteht als die ganzzahlige Stellfolge. Da die reellwertige Stellfolge ausschließlich virtuell berechnet wird, ist es bei ausreichender Rechenkapazität möglich, mehrere virtuelle Zyklen in einem tatsächlichen Zyklus zu realisieren. Andernfalls führt die Berücksichtigung der virtuellen Stellfolge zu einem deutlich langsameren Konvergenzverhalten. Dies gilt insbesondere in der Nähe der Lösung der bezüglich des Regelfehlers optimalen Steuerung.

Bezogen auf die Kriterien für eine ganzzahlige ILR ist für diesen Ansatz Folgendes festzuhalten:

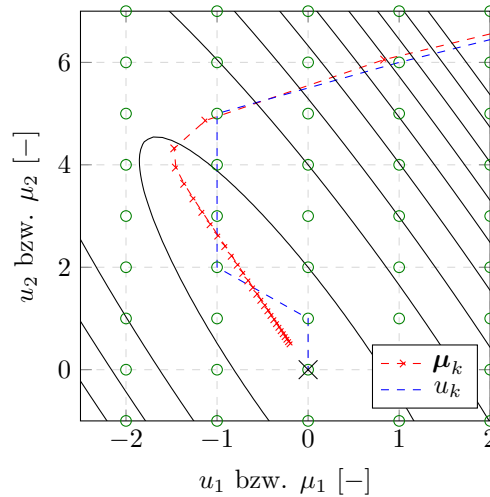


Abbildung 6.3: Konturplot der Kostenfunktion einer optimalen Steuerung, die sich aus der Lösung von QP 2 ergibt. Die zulässigen Lösungen sind mit grünen Kreisen markiert. Die roten Kreuze markieren den Verlauf der virtuellen, reellwertigen Stellfolge, die in der parallelen ILR berechnet wird. Die blaue Linie zeigt den Verlauf der tatsächlichen, ganzzahligen Stellfolge, die auf das System angewendet wird. Der zeitliche Zusammenhang beider Stellfolgen ist dieser Darstellung nicht zu entnehmen.

- **Fehlerschranke:** Wie in Schäpel et al., 2018 beschrieben, kann der Fall auftreten, dass sich die reellwertige und ganzzahlige Stellfolge von unterschiedlichen Seiten der optimalen Stellfolge bezogen auf den Fehler nähern. In einem solchen Fall ist es möglich, dass die reellwertige Stellfolge die Stellfolge der optimalen Steuerung aus QP 2 erreicht, ohne die Kostenfunktion der ganzzahligen Optimierung so zu verschieben, dass auch für den beschränkten Fall eine finale Anpassung möglich ist. Eine sichere Fehlerschranke kann daher nicht angegeben werden.
- **Monotone Konvergenz:** Da wie beschrieben die Verwendung der reellwertigen Stellfolge den linearen Parameter aus QP 3 beeinflusst, wird das Optimum auf unbekannte Weise verschoben. Dadurch gelten die bekannten Zusammenhänge zwischen den Zyklen nicht mehr zwangsläufig und monotone Konvergenz kann nicht mehr garantiert werden.
- **Deterministischer Rechenaufwand:** Prinzipiell gelten die gleichen Annahmen wie für den Fall eines adaptiven Gewichts wie er oben beschrieben wurde. Allerdings schwankt der tatsächliche Rechenaufwand der *Branch-and-Bound*-Suche nicht so stark, da der quadratische Parameter des QPs gleich bleibt und somit die Form des Optimierungsproblems weniger stark variiert.

6.3.2 Optimale-Menge-Synthese

In der Kostenfunktion aus Gl. (6.29) ist die Bestrafung der Änderung der Stellfolge $\Delta \mathbf{u}$ notwendig, um zu große Transienten im Lernfortschritt zu vermeiden, die das System in ungewollte Oszillationen versetzen können. Allerdings sorgt die Bestrafung im ganzzahligen Fall nicht für eine Verkleinerung des Lösungsraums, die den Rechenaufwand der Exploration des Suchbaums verbessern würde. Wenn die optimale Stellfolge für den nächsten Zyklus \mathbf{u}_{k+1} numerisch berechnet wird, ist daher eine explizite Beschränkung der Menge der zulässigen Stellfolgen sinnvoll. In diesem Fall kann auf die Bestrafung der Änderung der Stellfolge verzichtet werden.

Quadratisches Programm (QP) 4: Die Stellfolge für den Zyklus $k+1$ einer ganzzahligen ILR, in der die Änderung der Stellfolge explizit beschränkt ist, ergibt sich aus

$$\mathbf{u}_{k+1} = \arg \min_{\tilde{\mathbf{u}}_{k+1}} \tilde{\mathbf{u}}_{k+1}^T \mathbf{Q} \tilde{\mathbf{u}}_{k+1} + \mathbf{q}^T \tilde{\mathbf{u}}_{k+1} \quad (6.74)$$

mit

$$\mathbf{Q} = \mathbf{G}^T \mathbf{W}_E \mathbf{G} \quad (6.75)$$

$$\mathbf{q}^T = -2 \left(\mathbf{u}_k^T \mathbf{G}^T \mathbf{W}_E \mathbf{G} + \mathbf{e}_k^T \mathbf{W}_E \mathbf{G} \right) \quad (6.76)$$

unter der Beschränkung

$$\tilde{\mathbf{u}}_{k+1} \in \mathcal{S}_{k+1} \subsetneq \mathcal{U} \subsetneq \mathbb{N}^{n_u} \quad (6.77)$$

Damit muss für jede Iteration $k+1$ eine zyklusspezifische Menge \mathcal{S}_{k+1} definiert werden, die den zulässigen Lösungsraum in der entsprechenden Iteration beschreibt. Durch die Wahl der Menge \mathcal{S}_{k+1} kann so die maximale Anzahl der Explorationsschritte gezielt reduziert und damit der maximale Rechenaufwand einer *Branch-and-Bound*-Suche beschränkt werden.

Die Lösungsmenge

$$\mathcal{S}_{k+1} = \{\mathbf{u}_{k+1}^1, \dots, \mathbf{u}_{k+1}^i, \dots, \mathbf{u}_{k+1}^{n_{\mathcal{S},k+1}}\} \quad (6.78)$$

besteht aus einer definierten Anzahl $n_{\mathcal{S},k+1}$ an Lösungsoptionen \mathbf{u}_{k+1}^i . Für die potentiellen Lösungen gilt

$$\mathbf{u}_{k+1}^i = \mathbf{u}_k + \mathbf{s}_{k+1}^i \quad (6.79)$$

wobei \mathbf{s}_{k+1}^i einen zulässigen Schritt bzw. Änderung der Stellgröße zwischen zwei Zyklen beschreibt. Das Lernverhalten der ILR wird maßgeblich durch die Zusammensetzung der Menge \mathcal{S}_{k+1} und folglich den zulässigen Schritten \mathbf{s}_{k+1}^i beeinflusst. In Abb. 6.4 ist ein Beispiel für eine zweidimensionale Stellfolge gezeigt. Mit den in rot gezeigten Schritten ist von der ak-

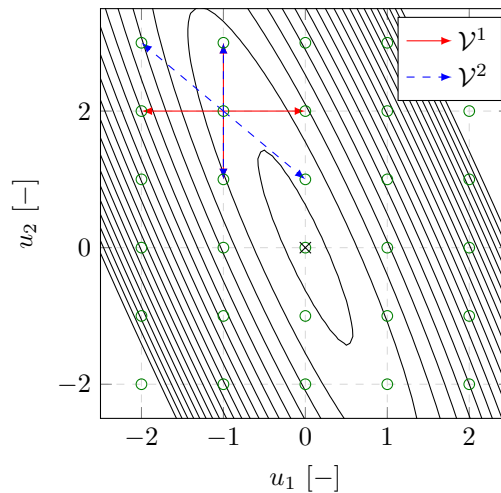


Abbildung 6.4: Konturplot einer Kostenfunktion. Alle zulässigen Lösungen sind mit einem grünen Kreis markiert. Ausgehend von einer Stellfolge \mathbf{u}_k sind zwei verschiedene Mengen an zulässigen Schritten gezeigt.

tuellen Stellfolge $\mathbf{u}_k = [-1, 3]^T$ keine Verbesserung mehr möglich, obwohl das Optimum der

Kostenfunktion offensichtlich nicht erreicht wurde. Im Gegensatz dazu würden die zulässigen Schritte, die in blau gezeigt sind, eine weitere Verbesserung der Stellfolge zulassen. Die zulässigen Schritte können somit als Entwurfsparameter für die Reglersynthese betrachtet werden.

Dafür wird zunächst angenommen, dass die zulässigen Schritte zyklusinvariant sind. Damit folgt aus der Menge der zulässigen n_S Schritte

$$\mathcal{V}_{k+1} = \mathcal{V}_k = \mathcal{V} = \{\mathbf{s}^1, \dots, \mathbf{s}^i, \dots, \mathbf{s}^{n_S}\} \quad (6.80)$$

und der Stellfolge des aktuellen Zyklus \mathbf{u}_k direkt die Lösungsmenge

$$\mathcal{S}_{k+1} = \{\mathbf{u}_k + \mathbf{s}^1, \dots, \mathbf{u}_k + \mathbf{s}^i, \dots, \mathbf{u}_k + \mathbf{s}^{n_S}\} \quad (6.81)$$

Für die Auswahl der Menge \mathcal{V} muss zunächst eine Kennzahl bestimmt werden, an der die Qualität der Menge bezogen auf die erreichbare Güte der Regelung bewertet wird. Dafür wird die Kostenfunktion für die konvergierte Stellfolge \mathbf{u}_N betrachtet.

Quadratisches Programm (QP) 5: Die bezüglich des Regelfehlers bestmögliche Lösung J^{opt} für einen Fall ohne eine Beschränkung der zulässigen Schritte auf eine Menge $\mathcal{S}_{k+1} \subsetneq \mathcal{U} \subsetneq \mathbb{N}^{n_u}$ ergibt sich aus der Lösung von

$$J^{\text{opt}} = \min_{\tilde{\mathbf{u}}_N} \tilde{\mathbf{u}}_N^T \mathbf{Q} \tilde{\mathbf{u}}_N + \mathbf{q}^T \tilde{\mathbf{u}}_N \quad (6.82)$$

unter der Bedingung

$$\mathcal{U} \subsetneq \mathbb{N}^{n_u} \quad (6.83)$$

mit \mathbf{Q} und \mathbf{q} gemäß Gl. (6.75) und Gl. (6.76).

Wie in Abschnitt 6.2.2 diskutiert, ist im beschränkten Fall die konvergierte Stellfolge \mathbf{u}_N nicht eindeutig, sondern hängt von der initialen Stellfolge \mathbf{u}_0 , dem entsprechenden Regelfehler \mathbf{e}_0 , der Referenz und möglichen Störungen ab. Daher kann im beschränkten Fall nur eine obere Grenze J^+ für die Kostenfunktion bestimmt werden.

Quadratisches Programm (QP) 6: Die obere Schranke für die erreichbare Güte der Regelung J^+ in Abhängigkeit einer Menge an zulässigen Schritten \mathcal{V} ergibt sich aus

$$J^+ = \max_{\tilde{\mathbf{u}}_N} \tilde{\mathbf{u}}_N^T \mathbf{Q} \tilde{\mathbf{u}}_N + \mathbf{q}^T \tilde{\mathbf{u}}_N \quad (6.84)$$

unter der Beschränkung

$$\tilde{\mathbf{u}}_N^T \mathbf{Q} \tilde{\mathbf{u}}_N + \mathbf{q}^T \tilde{\mathbf{u}}_N \leq \left(\tilde{\mathbf{u}}_N + \mathbf{s}^i \right)^T \mathbf{Q} \left(\tilde{\mathbf{u}}_N + \mathbf{s}^i \right) + \mathbf{q}^T \left(\tilde{\mathbf{u}}_N + \mathbf{s}^i \right) \quad \forall \mathbf{s}^i \in \mathcal{V} \quad (6.85)$$

$$\tilde{\mathbf{u}}_N, \tilde{\mathbf{u}}_N + \mathbf{s}^i \in \mathcal{U} \subsetneq \mathbb{N}^{n_u} \quad (6.86)$$

mit \mathbf{Q} und \mathbf{q} gemäß Gl. (6.75) und Gl. (6.76).

Welche Lösungen gemäß der Beschränkung aus Gl. (6.85) zulässig sind, ist in Abb. 6.5 beispielhaft illustriert. Es ist zu erkennen, dass von den Stellfolgen \mathbf{u}_N^1 , \mathbf{u}_N^2 kein Schritt aus der Menge \mathcal{V} eine bessere Lösung liefert. Im Gegensatz dazu ist für \mathbf{u}_N^3 eine Verbesserung mit Schritten aus der Menge \mathcal{V} möglich. Folglich sind nur \mathbf{u}_N^1 und \mathbf{u}_N^2 zulässige, optimierende Stellgrößen von Gl. (6.84) gemäß der Beschränkung Gl. (6.85).

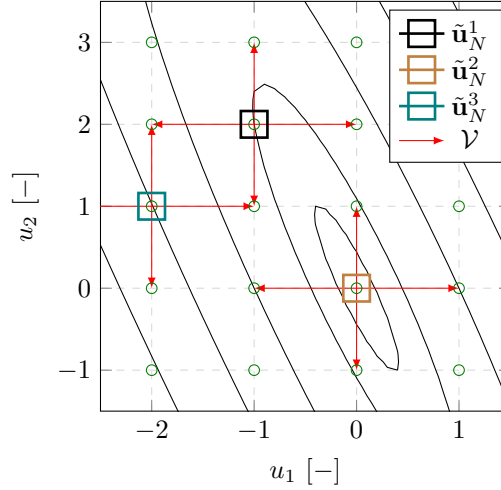


Abbildung 6.5: Konturplot einer Kostenfunktion. Alle zulässigen, optimierenden Lösungen sind mit einem grünen Kreis markiert. Exemplarisch sind die Stellfolgen \mathbf{u}_N^1 , \mathbf{u}_N^2 und \mathbf{u}_N^3 mit Vierecken markiert und die zulässigen Schritte einer Menge \mathcal{V} durch Pfeile visualisiert. \mathbf{u}_N^1 , \mathbf{u}_N^2 sind gemäß der Beschränkung Gl. (6.85) zulässige, optimierende Lösungen.

Anmerkung 8: Die Stellfolgen \mathbf{u}_N^1 und \mathbf{u}_N^2 legen folglich lokale Minima des beschränkten Optimierungsproblems fest.

Da die obere Schranke J^+ über die Beschränkungen von der Menge der zulässigen Schritte \mathcal{V} abhängt, kann QP 6 genutzt werden, um die Menge \mathcal{V} selbst zu optimieren. Ziel ist es, eine obere Schranke J^+ zu finden, die möglichst dicht an der bestmöglichen Performance J^{opt} liegt, sprich möglichst klein ist. Dafür kann ein nichtlineares Minimierungsproblem betrachtet werden, was auf QP 6 aufbaut.

Nichtlineares Programm (NLP) 1: Die optimale Menge \mathcal{V} kann über die Minimierung

$$\mathcal{V} = \arg \min_{\tilde{\mathcal{V}}} \left[\max_{\tilde{\mathbf{u}}_N} \tilde{\mathbf{u}}_N^T \mathbf{Q} \tilde{\mathbf{u}}_N + \mathbf{q}^T \tilde{\mathbf{u}}_N \right] \quad (6.87)$$

unter der Bedingung

$$\tilde{\mathbf{u}}_N^T \mathbf{Q} \tilde{\mathbf{u}}_N + \mathbf{q}^T \tilde{\mathbf{u}}_N \leq \left(\tilde{\mathbf{u}}_N + \mathbf{s}^i \right)^T \mathbf{Q} \left(\tilde{\mathbf{u}}_N + \mathbf{s}^i \right) + \mathbf{q}^T \left(\tilde{\mathbf{u}}_N + \mathbf{s}^i \right) \quad \forall \mathbf{s}^i \in \tilde{\mathcal{V}} \quad (6.88)$$

$$\tilde{\mathbf{u}}_N, \tilde{\mathbf{u}}_N + \mathbf{s}^i \in \mathcal{U} \subsetneq \mathbb{N}^{n_u} \quad (6.89)$$

$$|\tilde{\mathcal{V}}| = n_S \quad (6.90)$$

mit \mathbf{Q} und \mathbf{q} gemäß Gl. (6.75) und Gl. (6.76) bestimmt werden.

Anmerkung 9: Gleichung 6.90 beschränkt die Kardinalität der Menge \mathcal{V} auf einen festen Wert n_S . Dadurch kann die Menge \mathcal{V} für die Implementierung durch einen Supervektor $[\mathbf{s}^{1T}, \mathbf{s}^{2T}, \dots, \mathbf{s}^{n_S T}]^T$ repräsentiert werden.

Die Lösung des NLP 1 erfordert zwar einen hohen Rechenaufwand, allerdings kann es vor der eigentlichen Verwendung der Regelung gelöst werden. Für die Bestimmung einer optimalen Menge \mathcal{V} ist lediglich die Messung des Regelfehlers \mathbf{e}_0 in Abhängigkeit einer beliebigen Aktuierung \mathbf{u}_0 erforderlich, um \mathbf{Q} und \mathbf{q} gemäß Gl. (6.75) und Gl. (6.76) zu bestimmen. Diese Messung kann vor dem eigentlichen Betrieb durchgeführt werden, ohne dass dafür eine Regelung implementiert sein muss.

6.3.3 Simulationsstudie für ein generisches Beispiel

Um die Unterschiede in den einzelnen Syntheseverfahren zu verdeutlichen, soll im Folgenden das Verhalten einer ganzzahligen ILR an einem einfachen, generischen Simulationsbeispiel aufgezeigt werden. Dafür wird ein schwingungsfähiges System mit der Übertragungsfunktion

$$G(s) = \frac{1}{\frac{1}{\omega_0^2} s^2 + \frac{2D}{\omega_0} s + 1} \quad (6.91)$$

angenommen. Die Eigenkreisfrequenz wird in diesem Beispiel zu $\omega_0 = 2\pi \cdot 50 \text{ rad/s}$ gewählt und der Dämpfungsfaktor als $D = 0.2$ festgelegt. Ziel der ILR ist es, einer sinusförmigen Referenz

$$r(t) = A \cdot \sin(f_r \cdot 2\pi \cdot t) \quad (6.92)$$

möglichst optimal zu folgen. Dabei wird eine Amplitude von $A = 5$ und eine Frequenz von $f_r = 10 \text{ Hz}$ für die Referenz angenommen. Für die Aktuierung gilt in diesem Beispiel eine Beschränkung auf einen 4-bit-Raum, was 16 diskreten Stellmöglichkeiten entspricht. Mit einer Aktuationsfrequenz von $f_u = 200 \text{ Hz}$ ergibt sich eine Dimension der Stellgröße von $n_u = 20$. Die Abtastung des Systemausgangs erfolgt mit $f_y = 1 \text{ kHz}$.¹

Im Folgenden werden die beiden Ad-hoc-Ansätze mit adaptivem Gewicht bzw. paralleler, reellwertiger ILR gegen das vorgestellte Syntheseverfahren auf Basis einer optimalen Menge zulässiger Schritte verglichen. Letzteres wird im Folgenden als OMS bezeichnet. Für die ersten beiden Ansätze wird ein initiales Verhältnis von

$$\mathbf{W}_{\Delta U}^0 = 100 \cdot \mathbf{W}_E \quad (6.93)$$

für die Gewichte der Kostenfunktion gewählt. Die Anpassung des Gewichts $\mathbf{W}_{\Delta U}$ erfolgt mit einem Adaptionfaktor von $p_a = 0.95$ über die Adaptionsregel

$$\mathbf{W}_{\Delta U}^{k+1} = \begin{cases} p_a \cdot \mathbf{W}_{\Delta U}^k & \text{falls } \|\mathbf{u}_k - \mathbf{u}_{k-1}\|_2 = 0 \\ \frac{1}{p_a} \cdot \mathbf{W}_{\Delta U}^k & \text{falls } \|\mathbf{u}_k - \mathbf{u}_{k-1}\|_2 \neq 0, \|\mathbf{u}_{k-1} - \mathbf{u}_{k-2}\|_2 = 0 \\ \mathbf{W}_{\Delta U}^k & \text{sonst.} \end{cases} \quad (6.94)$$

Der zweite Fall der Adaptionsregel stellt sicher, dass das Gewicht nicht beliebig klein wird, wenn keine Änderung auftritt. Damit ist sichergestellt, dass die Regelung bei unerwarteten Störungen keine ungewollten Oszillationen hervorruft. Die OMS wird mit einer Kardinalität von $n_S = 100$ durchgeführt. Um die Vorteile der OMS hervorzuheben, werden zum Vergleich noch zusätzlich die Lösungen mit einer generischen, nicht optimierten Menge \mathcal{V} aus $n_u = 20$ Einheitsvektoren dargestellt.

Abbildung 6.6 zeigt auf der linken Seite die Entwicklung der 2-Norm des Fehlers über die Zyklen. Es wurde in allen Fällen eine initiale Stellfolge $\mathbf{u}_0 = \mathbf{0}$ angenommen. Wie in der theoretischen Einführung diskutiert, erreicht der Ansatz mit einem adaptiven Gewicht die Stellfolge, die bezüglich des Fehlers optimal ist. Im Gegensatz dazu kann mit dem Ansatz einer parallelen ILR diese optimale Stellfolge nicht erreicht werden. Zudem ist hier deutlich das nicht monotone Lernverhalten zu erkennen. Die ILR, die über die OMS ausgelegt wurde, erreicht in diesem Beispiel ebenfalls die optimale Stellfolge. Der Vorteil der OMS wird vor allem im direkten Vergleich mit der generischen Menge deutlich. Als obere Grenze gemäß

¹Es wurde an dieser Stelle eine höhere Abtastung der Messung gewählt, um die Systemdynamik exakter bei der Berechnung der Stellfolge zu berücksichtigen. So kann ein besseres Folgeverhalten erzielt werden.

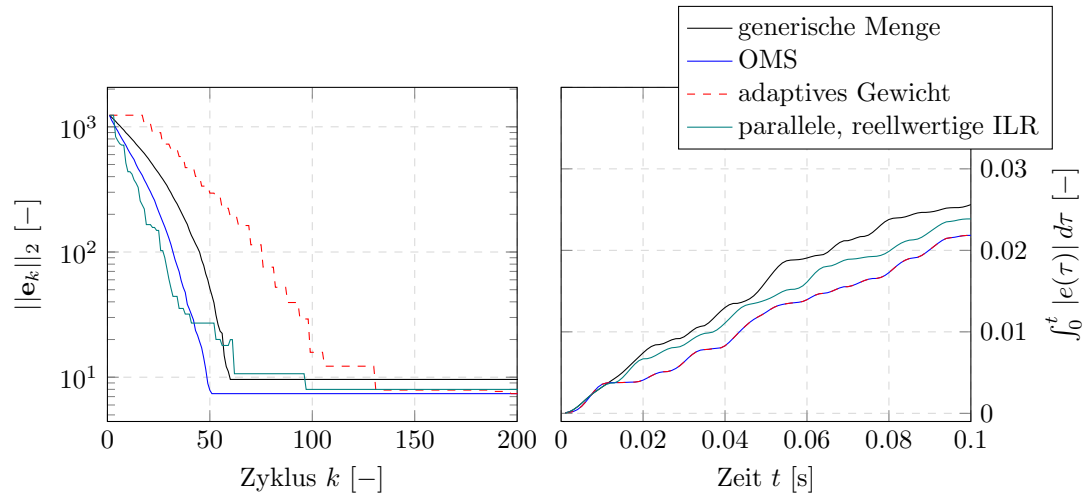


Abbildung 6.6: Ergebnisse für verschiedene ILR-Synthesemethoden ausgehend von einer initialen Stellfolge $\mathbf{u}_0 = \mathbf{0}$.

Links: Entwicklung der 2-Norm des Fehlers über die Zyklen.

Rechts: Aufintegrierter Regelfehler über einen Zyklus, der aus der konvergierten Stellfolge resultiert.

QP 6 für die optimale Menge ergibt sich $J_{\mathcal{V}}^+ = -1205$ und für die generische Menge aus Einheitsvektoren $J_g^+ = -1050$. Damit kann der Unterschied des Regelfehlers im konvergierten Zustand in einem ungünstigen Fall sehr deutlich sein. In diesem konkreten Beispiel liegen die Werte der Kostenfunktion im konvergierten Zustand näher beieinander. Es ergeben sich Werte von $J_{\mathcal{V}} = -1230$ bzw. $J_g = -1227$. Die Konvergenzgeschwindigkeit der Ansätze mit adaptivem Gewicht und paralleler, reellwertiger ILR kann durch eine andere Wahl des Gewichts $\mathbf{W}_{\Delta U}$ angepasst werden. Um die Auswirkungen der unterschiedlichen, konvergierten Stellfolgen zu illustrieren, ist auf der rechten Seite von Abb. 6.6 der integrierte Fehler dargestellt, der sich bei einer jeweiligen Aktuation mit den konvergierten Stellfolgen ergibt. Hier zeigt sich deutlich der geringere Fehler, der aus der optimalen Stellfolge resultiert. Die zugehörigen, konvergierten Stellfolgen sind Abb. 6.7 gezeigt. Auch hier zeigt sich ein deutlicher Unterschied.

Wie in Abschnitt 6.2.2 diskutiert, ist die Performance der Ansätze abhängig von der initialen Stellfolge \mathbf{u}_0 . Dies gilt insbesondere für den Ansatz der OMS. Um dies zu veranschaulichen, ist in Abb. 6.8 noch einmal die Entwicklung der Fehlernorm über die Zyklen auf der linken Seite dargestellt und auf der rechten Seite wiederum der integrierte Fehler über einen Zyklus bei der Anwendung der konvergierten Stellfolgen. Für diesen Fall wurde allerdings eine zufällige initiale Stellfolge $\mathbf{u}_0 \neq \mathbf{0}$ gewählt.

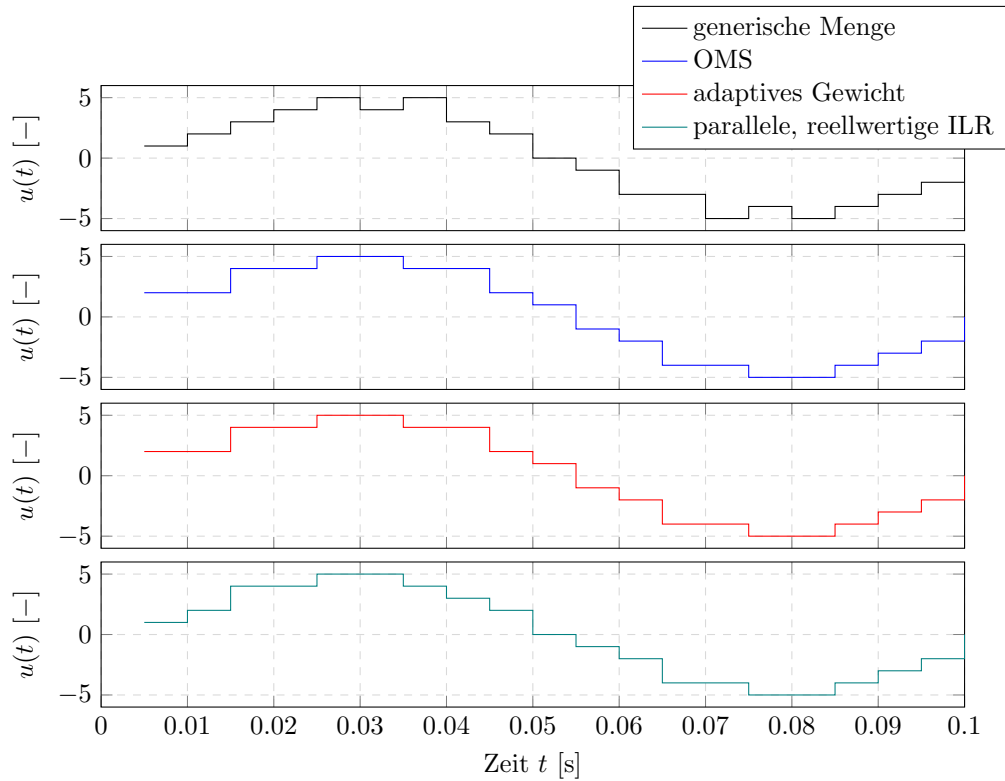


Abbildung 6.7: Konvergierte ILR-Stellfolgen für einen Zyklus, die sich aus einer Synthese mit adaptivem Gewicht, paralleler, reellwertiger ILR und der OMS ergeben. Zusätzlich ist ein Ergebnis mit einer generischen Menge gezeigt.

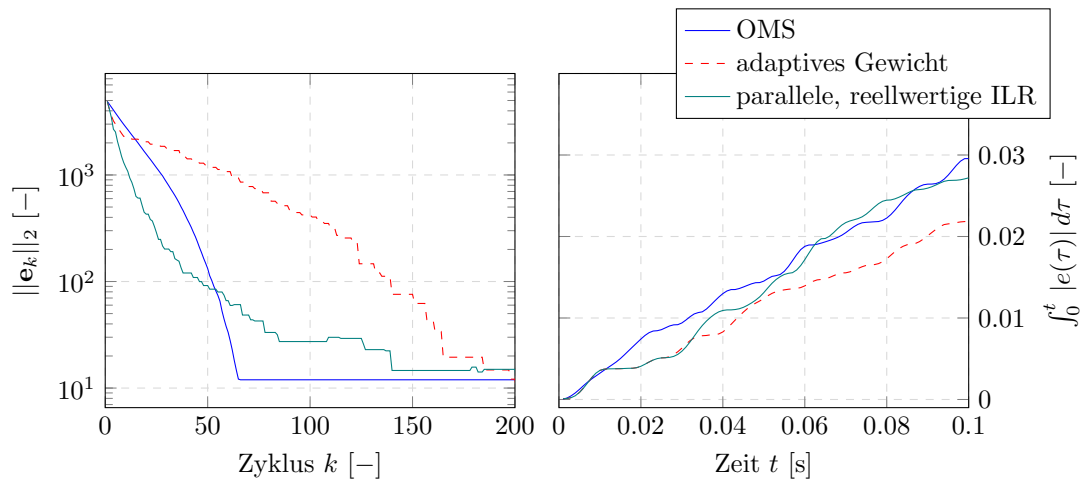


Abbildung 6.8: Ergebnisse für verschiedene ILR Synthesemethoden ausgehend von einer zufälligen initialen Stellfolge $\mathbf{u}_0 \neq \mathbf{0}$.

Links: Entwicklung der 2-Norm des Fehlers über die Zyklen.

Rechts: Aufintegrierter Regelfehler über einen Zyklus, der aus der konvergierten Stellfolge resultiert.

Hier zeigt sich, dass mit der OMS, im Gegensatz zum Ansatz mit einem adaptiven Gewicht, für diese Anfangsbedingung ein abweichendes Ergebnis im Bezug auf den absoluten Regelfehler erreicht wird. Dennoch zeigt sich auch hier wieder das gewünschte monotone Konvergenzverhalten. Zudem wird nach wie vor die ermittelte obere Grenze von $J_V^+ = -1205$

eingehalten. Der Ansatz mit einer parallelen, reellwertigen ILR zeigt auch hier die bereits diskutierten Probleme im Konvergenzverhalten. Die konvergierten Stellfolgen der drei Ansätze sind Abb. 6.9 dargestellt. Hier zeigt sich nun deutlich der Unterschied zwischen der auf

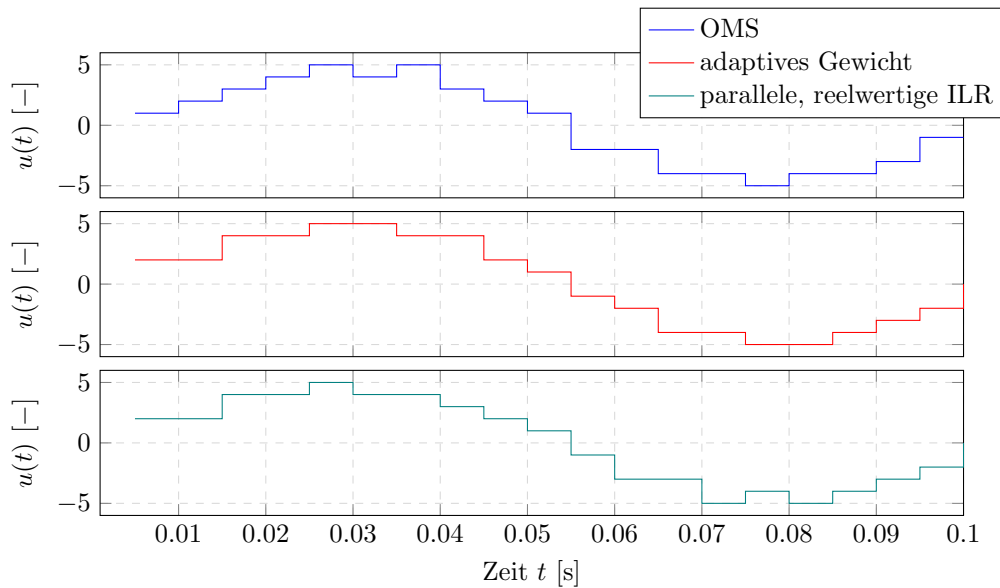


Abbildung 6.9: Konvergierte ILR-Stellfolgen für einen Zyklus, die sich aus einer Synthese mit adaptivem Gewicht, paralleler, reellwertiger ILR und der OMS ergeben.

den Regelfehler bezogenen optimalen Stellfolge, die sich mit dem Ansatz über ein adaptives Gewicht ergibt, und dem Ansatz der OMS.

Die Ergebnisse heben hervor, dass die OMS das in den Kriterien aus Abschnitt 6.3 geforderte Konvergenzverhalten zeigt. Der größte Vorteil der Methode ist allerdings der Rechenaufwand. Wie bereits oben diskutiert, benötigt eine ganzzahlige ILR, die mit der OMS ausgelegt wurde, deterministischen Rechenaufwand in jeder Iteration, da nur eine sehr kleine Menge an Lösungen ausgewertet werden muss. Damit ist die Regelung im Gegensatz zu den anderen Ansätzen problemlos in Systemen mit harten Echtzeitanforderungen verwendbar. Da die Ansätze auf verschiedenen Methodiken beruhen, ist es schwierig, eine Kenngröße für den Rechenaufwand der Algorithmen anzugeben. Um trotzdem einen repräsentativen Vergleich zu ermöglichen, wird im Folgenden die Anzahl der Simplex-Iterationen für die Methoden gewählt, die innerhalb der „*Branch-and-Bound*“-Suche den gesamten Suchraum betrachten, und die Anzahl der Auswertungen der Kostenfunktion, die für eine mit der OMS ausgelegte Regelung maximal notwendig ist. In Abb. 6.10 sind die jeweiligen Werte über die Zyklen aufgetragen. Hier ist zu erkennen, dass der Rechenaufwand für die beiden Methoden mit vollständiger *Branch-and-Bound*-Suche sehr stark variiert. Vor allem bei einer Reduzierung des Gewichts $\mathbf{W}_{\Delta U}$ im adaptiven Ansatz steigt der Rechenaufwand stark an und liegt teilweise bis zu zwei Größenordnungen über dem Rechenaufwand für die ILR aus der OMS. Ein großes Gewicht $\mathbf{W}_{\Delta U}$ führt zu einer Regularisierung über die Diagonale des quadratischen Parameters \mathbf{Q} und damit zu einer schnelleren Lösung. Durch die Anpassung des Gewichts unterliegt der Aufwand über die Iterationen starken Schwankungen. Auch der Ansatz mit einer parallelen, reellwertigen ILR erfordert einen schwankenden Rechenaufwand über die Iterationen, der zudem ebenfalls deutlich über dem Ansatz der optimalen Menge liegt. Beide Verfahren mit *Branch-and-Bound*-Suche liegen im Schnitt über eine Größenordnung über der Regelung aus der OMS.

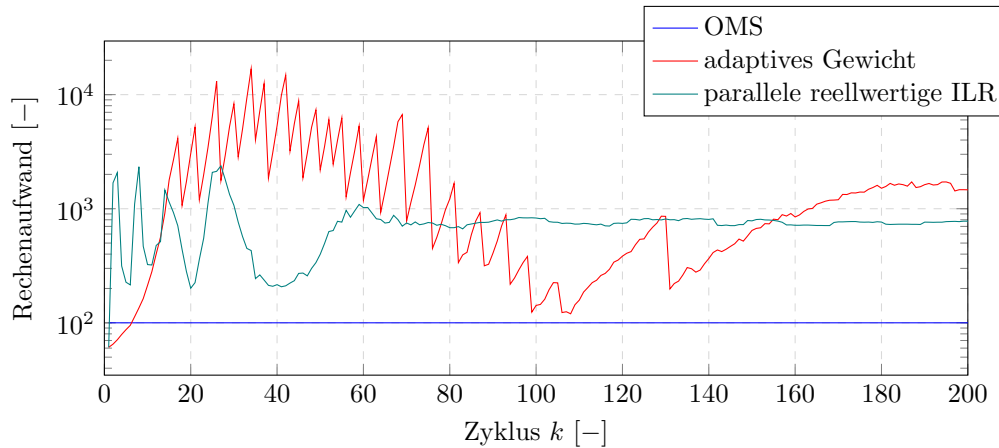


Abbildung 6.10: Rechenaufwand verschiedener ILR-Verfahren über die Zyklen. Der Rechenaufwand wird an der Anzahl der Simplex-Iterationen (adaptives Gewicht, parallele ILR) bzw. der maximale Anzahl der Auswertungen der Kostenfunktion (OMS) bemessen.

6.4 Binäre iterativ lernende Regelung

Ein Sonderfall der ganzzahligen ILR ist die Beschränkung der Stellgröße \mathbf{u}_k auf einen binären Wertebereich. Dies ist vor allem für schaltende Systeme sinnvoll, in denen einzelne Schaltelemente separat angesteuert werden sollen oder auch für allgemeine Entscheidungssysteme. Die Herausforderung bei dieser Art von System ist oft die große Dimension der Stellgröße, wenn ein System aus mehreren Schalt- bzw. Entscheidungselementen besteht und/oder eine hohe Aktuationsfrequenz notwendig ist. Vor allem hohe Aktuationsfrequenzen für binäre Systeme sind üblich, um die geringe Auflösung der Amplitude der Stellgröße zu kompensieren. In diesem Abschnitt wird anhand eines generischen Beispiels noch einmal aufgezeigt, unter welchen Bedingungen mit einer ganzzahligen Optimierung als Grundlage bessere Ergebnisse erzielt werden als mit einfachen Rundungsansätzen. Im Anschluss wird dann ein binärer ILR-Ansatz näher vorgestellt, der für die Synchronisation der Verbrennungsrohre in einer Gasturbine mit einer Gleichraumverbrennung genutzt werden kann. Dafür wird zuvor eine Methode eingeführt, die den Rechenaufwand für Systeme mit einer hochdimensionalen Stellfolge unter bestimmten Umständen deutlich reduzieren kann.

6.4.1 Vergleich mit Rundungsansatz

Wie bereits in der Einleitung diskutiert, ermöglicht die Optimierung im ganzzahligen Wertebereich unter bestimmten Umständen eine deutlich bessere Güte der Regelung als eine einfache Rundung. In diesem Abschnitt soll nochmal an einem konkreten Beispiel für den Sonderfall einer binären Aktuation aufgezeigt werden, für welche Systeme die Integration der Beschränkung in die Optimierung besonders vorteilhaft ist, um gutes Folgeverhalten zu erzielen. Dafür wird der Vergleich zu einer bezüglich der Differenz zwischen ganzzahliger und reellwertiger Stellfolge optimierten Rundungsstrategie, dem sogenannten *Sum-Up-Rounding* (SUR), gezogen. Dieser Vergleich wurde auch in Arnold, Neuhäuser und King, 2020 veröffentlicht.

SUR ist ein in Sager, Bock und Diehl, 2007 vorgestellter Rundungsansatz, der eine kontinuierliche Stellfolge $\boldsymbol{\mu}(t)$ in eine binäre Stellfolge $\mathbf{u}(t)$ übersetzt und dabei eine Grenze für die

integrale Differenz

$$\int_0^t \mu(\tau) - u(\tau) d\tau \quad (6.95)$$

der Aktuation über der Zeit garantiert. Einer der in Sager et al., 2007 aufgeführten Vorteile des Ansatzes ist, dass kein festes Zeitraster angenommen werden muss, womit ein Schalten zu beliebigen Zeiten möglich ist. In diesem Fall zeigt sich ein deutlicher Unterschied zur konventionellen Pulsweitenmodulation (PWM). Da in einer realen Applikation allerdings immer eine diskrete äquidistante Zeitabtastung notwendig ist, wird hier nur dieser Fall diskutiert. Für weitere Details der Methode sei auf Sager et al., 2007 verwiesen. Für den diskreten Fall verhält sich der Ansatz sehr ähnlich zu einer PWM. Mit der Schaltzeit $T_s = 1/f_u$ in Abhängigkeit der Aktuationsfrequenz f_u und der Periodendauer T_p des zyklischen Prozesses kann die Transformationsregel

$$u_i = \begin{cases} 1 & \text{falls } \left(\sum_{l=1}^i \mu_l - \sum_{l=1}^{i-1} u_l \right) T_s \geq 0.5 \cdot T_s \\ 0 & \text{sonst} \end{cases} \quad (6.96)$$

formuliert werden, die eine obere Grenze

$$\left\| \sum_{l=1}^{n_u} \mu_l - \sum_{l=1}^{n_u} u_l \right\|_{\infty} \leq 0.5 \quad (6.97)$$

definiert.

Ausgangspunkt für den Vergleich ist wieder ein System mit Verzögerung zweiter Ordnung mit der Übertragungsfunktion

$$G(s) = \frac{1}{\frac{1}{\omega_0^2} s^2 + 2\frac{D}{\omega_0} s + 1}. \quad (6.98)$$

Dabei ist $D \in [0, 1]$ die Dämpfungskonstante des Systems und $\omega_0 \in \mathbb{R}$ die Eigenkreisfrequenz. Für das Beispiel wird eine Abtastung des Systemausgangs mit $f_y = 1$ kHz angenommen. Die Aktuationsfrequenz wird zunächst auf $f_u = 200$ Hz festgelegt und später variiert. Wird die Aktuation konstant auf die Frequenz f_y interpoliert, kann das System nach einer Überführung in ein diskretes Zustandsraummodell entsprechend der Beschreibung in Abschnitt 6.1 in ein Eingangs-/Ausgangsmodell

$$\mathbf{y}_k = \mathbf{G}\boldsymbol{\mu}_k + \mathbf{b} \quad (6.99)$$

für den Zyklus k überführt werden. Für dieses Beispiel wird der Einfluss der Anfangsbedingung vernachlässigt. Für beide ILR-Ansätze werden die Gewichte im Verhältnis

$$\mathbf{W}_{\Delta U} = \mathbf{W}_E \quad (6.100)$$

gewählt und $\mathbf{W}_{\Delta U}$ wird nicht adaptiert. Für die ILR auf Basis der ganzzahligen Optimierung wurde kein spezielles Syntheseverfahren zur Verbesserung des Konvergenzverhaltens, wie sie in Abschnitt 6.3 beschrieben sind, verwendet, da sich in diesem Beispiel keine Problematik bei der Konvergenz gezeigt hat.

Für einen ersten Eindruck des Folgeverhaltens zeigt Abb. 6.11 den Systemausgang in Abhängigkeit der auskonvergierten Stellfolge, die sich durch eine ILR mit Rundungsansatz bzw. binärer Optimierung ergeben. Zusätzlich ist der optimale Verlauf der Regelgröße gezeigt, der

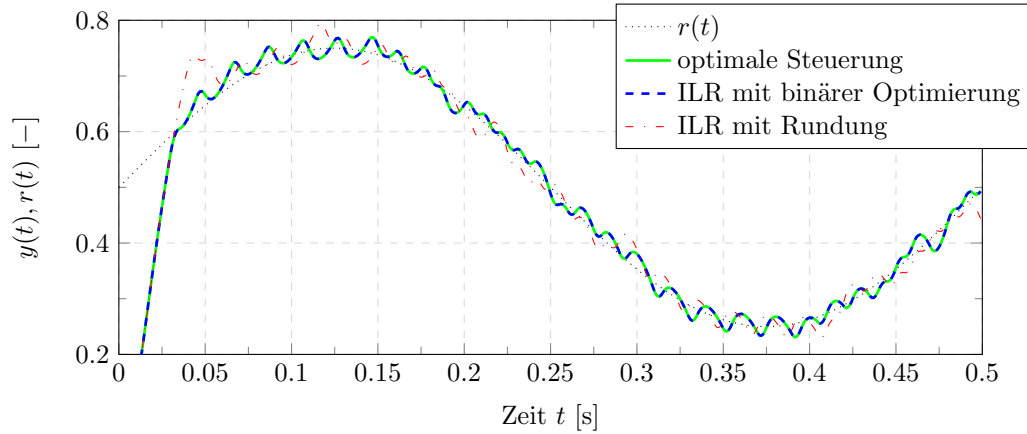


Abbildung 6.11: Folgeverhalten bezüglich einer Referenz $r(t)$. Gezeigt ist der Systemausgang bei Anwendung einer ganzzahligen ILR mit binärer Optimierung bzw. Rundungsverfahren und einer optimalen Steuerung. Die Dämpfung des Systems ist $D = 0.5$.

sich durch eine optimale binäre Steuerung ergeben würde. Es wurde zunächst eine Dämpfung von $D = 0.5$ eingestellt. In diesem einfachen Beispiel ohne zusätzliche Störungen oder Rauschen ist ersichtlich, dass der Ansatz auf Basis einer binären Optimierung im Gegensatz zum Rundungsansatz zur optimalen Lösung konvergiert. Trotzdem zeigt der Verlauf des Regelfehlers in Abb. 6.12, dass das Folgeverhalten des Rundungsansatz eine nur minimal schlechtere Qualität zeigt. Zudem ist ersichtlich, dass beide Ansätze erwartungsgemäß deutlich schlech-

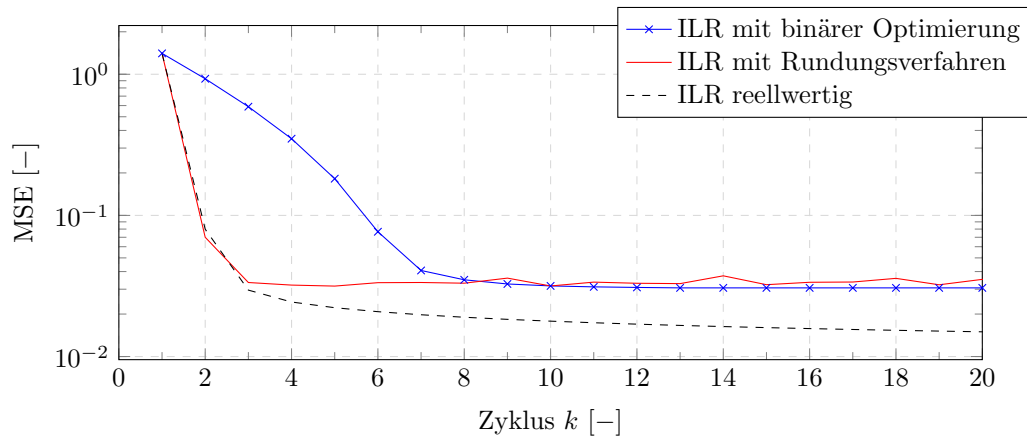


Abbildung 6.12: Mittlerer quadratischer Fehler (*Mean Squared Error*, MSE) über die Zyklen bei Verwendung einer ganzzahligen ILR mit binärer Optimierung bzw. Rundungsverfahren. Zusätzlich ist der Verlauf einer reellwertigen ILR dargestellt.

tere Ergebnisse liefern als eine reellwertige ILR.

Neben dem schlechteren Folgeverhalten kann das Schalten auch ungewollte Interaktionen mit der Systemdynamik hervorrufen. Die einzelnen Schaltvorgänge können ein schwingungsfähiges System möglicherweise in unerwünschter Weise anregen und damit Einfluss auf das Konvergenzverhalten der ILR haben. In Abb. 6.13 sind die Verläufe der Regelfehler über die Zyklen jeweils für ein System mit Dämpfung $D = 0.5$ bzw. $D = 0.1$ gezeigt. Es ist zu erkennen, dass für weniger stark gedämpfte Systeme die Stellfolge, die mit dem Rundungsansatz erhalten wird, schlechteres Konvergenzverhalten zeigt. Die Stellfolge konvergiert nicht vollständig, sondern oszilliert leicht für die Dämpfung $D = 0.5$ bzw. stark für $D = 0.1$. Im letzteren Fall

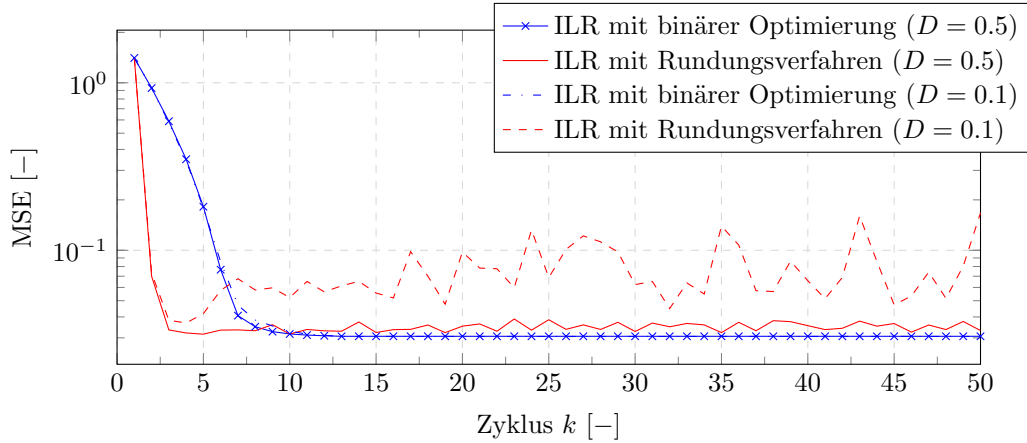


Abbildung 6.13: Mittlerer quadratischer Fehler (*Mean Squared Error*, MSE) über die Zyklen bei Verwendung einer ganzzahligen ILR mit binärer Optimierung bzw. Rundungsverfahren. Gezeigt sind jeweils die Ergebnisse für ein System mit einem Dämpfungsfaktor von $D = 0.5$ und $D = 0.1$.

scheint die Oszillation sogar über den Verlauf der Zyklen zuzunehmen. Der Ansatz mit einer binären Optimierung hingegen garantiert unabhängig von der Dämpfung die gleiche Qualität des Folgeverhaltens. In diesem Fall scheint es mit der vorgeschalteten, reellwertigen ILR nicht zu gelingen, die Rundungsvorschrift zu kompensieren bzw. deren Auswirkung zu erlernen.

Der Einfluss der Schaltfrequenz ist beispielhaft in Abb. 6.14 illustriert. Dabei wurden im Vergleich zu den bisherigen Beispielen geringere Schaltfrequenzen von $f_u = 100$ Hz bzw. $f_u = 40$ Hz angenommen. Erwartungsgemäß wird durch eine geringere Aktuationsfrequenz

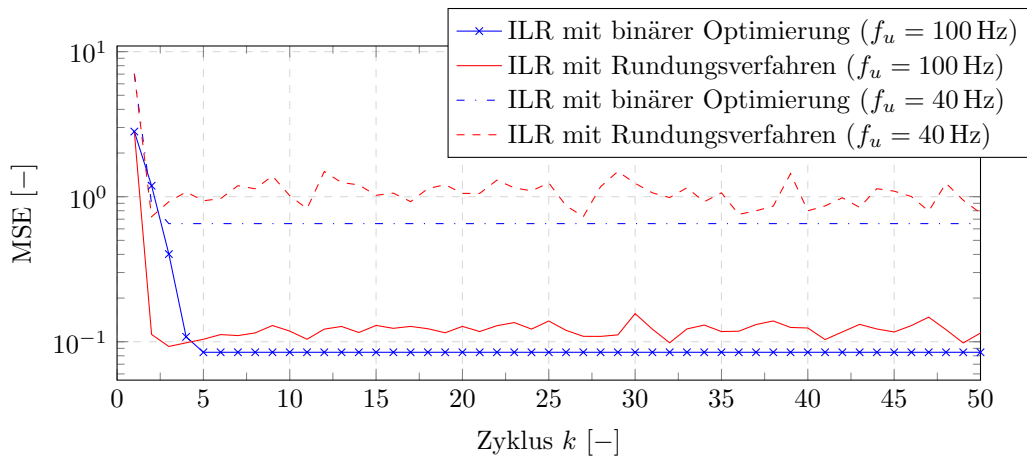


Abbildung 6.14: Mittlerer quadratischer Fehler (*Mean Squared Error*, MSE) über die Zyklen bei Verwendung einer ganzzahligen ILR mit binärer Optimierung bzw. Rundungsverfahren. Gezeigt sind jeweils die Ergebnisse für ein System mit einer Aktuationsfrequenz von $f_u = 100$ Hz ($n_u = 50$) und $f_u = 40$ Hz ($n_u = 20$).

das Niveau des Folgeverhaltens deutlich verschlechtert. Mit Blick auf die logarithmische Skalierung des Graphen ist jedoch zu erkennen, dass eine geringere Aktuationsfrequenz den Vorteil des Ansatzes mit binärer Optimierung deutlich vergrößert. Auch die Oszillationen des Rundungsansatzes nehmen mit größeren Schaltzeiten zu. Der Einfluss bezüglich der Schaltzeiten konnte in Arnold, Neuhäuser und King, 2020 auch an einem experimentellen Regelungsbeispiel am Aufbau einer linearen Statorkaskade bestätigt werden. Auch hier zeigte sich, dass für geringe Schaltzeiten der Ansatz mit einer binären Optimierung bessere Ergebnisse liefert als der Rundungsansatz.

Insgesamt lassen die gezeigten Ergebnisse darauf schließen, dass vor allem für Systeme mit unbekanntem oder sehr stark schwingendem Verhalten der Ansatz mit einer binären Optimierung dem Rundungsansatz vorzuziehen ist. Nur so kann sichergestellt werden, dass die Modellinformation direkt in die Berechnung der Stellgröße eingeht. Wenn Oszillationen ausgeschlossen werden können, bietet der Rundungsansatz jedoch eine sehr einfache und recheneffiziente Möglichkeit, binäre Stellgrößen zu realisieren. Dabei sollte jedoch sichergestellt werden, dass ein ausreichend schnelles Schalten des Aktuationselements möglich ist, um den Abstand gegenüber der optimalen Lösung klein zu halten. Allerdings muss festgehalten werden, dass Aussagen bezüglich der auf den Regelfehler bezogenen Optimalität des Rundungsansatzes in keinem Fall möglich sind.

6.5 Anwendung Feuersynchronisation

Im Folgenden soll ein spezielles Beispiel der Umsetzung einer binären ILR gegeben werden, die für die in der Einleitung erwähnte Synchronisation der Verbrennungsrohre in einer Gasturbine mit Gleichraumverbrennung verwendet werden kann. Dabei wird vor allem auf eine Methode zur Rechenzeitoptimierung einer binären ILR eingegangen, die notwendig ist, um das hochdimensionale Optimierungsproblem in ein Ersatzproblem zu übersetzen, welches in Echtzeit lösbar ist. Zunächst wird jedoch kurz auf den experimentellen Aufbau eingegangen.

6.5.1 Akustischer Plenumsprüfstand

Wie in der Einleitung beschrieben, ist eine der Herausforderungen beim Aufbau einer Gasturbine mit Gleichraumverbrennung die Kompensation von Druckschwankungen hinter den Verbrennungsrohren. Zwar ist zu erwarten, dass ein nachgeschaltetes Plenum an sich schon zu einer Dämpfung führt, für eine zufriedenstellende Kompensation der Druckschwankungen ist jedoch eine sinnvolle Synchronisation der Verbrennungsrohre notwendig. Durch eine geschickte Anpassung der Zündreihenfolge der einzelnen Rohre ist es möglich, das Druckfeld im Plenum zu beeinflussen und so in einer gewünschten Weise anzupassen.

Da ein Aufbau mit tatsächlichen Verbrennungsrohren mit einem enormen technischen Aufwand verbunden ist, wurde im Rahmen des Sonderforschungsbereich 1029 zunächst ein akustischer Ersatzprüfstand aufgebaut, um die benötigten Regelungskonzepte zu testen. Damit ist es möglich, bereits vor der eigentlichen Anwendung in einem Aufbau mit Verbrennungsrohren eine Regelungsstrategie zu entwickeln, welche die hohen Anforderungen an Geschwindigkeit und Recheneffizienz erfüllt. Es ist zwar davon auszugehen, dass in einem realen Aufbau das Druckfeld nicht mehr vollständig linear beschrieben werden kann, allerdings ist zu erwarten, dass wichtige Eigenschaften wie die Superposition von Drucksignaturen nach wie vor in guter Näherung gelten (Wolff, 2018).

Der genaue Aufbau des akustischen Plenumsversuchs ist ausführlich in Wolff, 2018 beschrieben. Der Aufbau wurde in diversen Arbeiten verwendet (Wolff & King, 2015; Wolff, Schäpel & King, 2016; Arnold, Topalovic & King, 2020). Da es an dieser Stelle ausschließlich um das regelungstechnische Konzept geht, wird hier nur auf den groben Aufbau bzw. den Modellierungsansatz eingegangen. Für technische Details sei auf die entsprechenden Ausführungen in Wolff, 2018 verwiesen. Eine Skizze des Aufbaus ist in Abb. 6.15 gezeigt. Das Plenum wurde in diesem Aufbau als Ringspalt realisiert, an dessen Kopfseite ein akustisch fester Rand angebracht ist, um den Übergang auf eine Turbine zu imitieren. Auf der anderen Seite sind

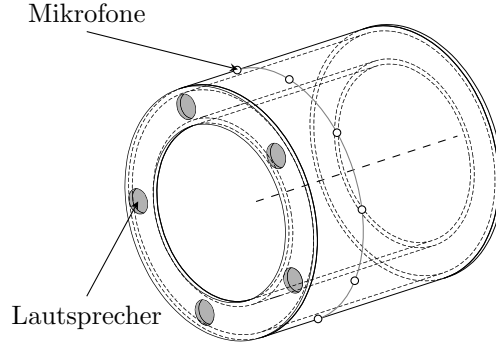


Abbildung 6.15: Skizze des akustischen Plenumsversuchs aus Wolff, 2018

fünf Lautsprecher angebracht, welche genutzt werden können, um die akustische Signatur einer Detonation abzuspielen und somit das akustische Feld im Plenum zu beeinflussen. Das akustische Feld wird über zehn Mikrofone am äußeren Umfang des Ringspalts gemessen.

Modellierung

Zur Beschreibung der Systemdynamik wird das in Wolff, 2018 beschriebene Modell aus Impulsantworten verwendet. Dabei werden die n_L Lautsprecher einzeln zyklisch bei einer Frequenz $f_f = 20$ Hz mit einem Detonationssignal angeregt. Der Beginn des Detonationssignals wird im Folgenden als Zündzeitpunkt bezeichnet. Die Ansteuerung der Lautsprecher im diskreten Zeitschritt l kann durch den binären Vektor

$$\boldsymbol{\nu}_l = [u_{l,1}, u_{l,2}, u_{l,3}, u_{l,4}, u_{l,5}]^T \quad (6.101)$$

dargestellt werden. Dabei bedeutet ein Wert $u_{l,i} = 1$, dass im Zeitschritt l des Zyklus begonnen wird, das Detonationssignal auf Lautsprecher i abzuspielen. In einem Aufbau mit Verbrennungsrohren würde dieser Zeitschritt dem Moment der Zündung entsprechen. Da jedes Rohr nur einmal pro Zyklus gefeuert werden soll, gibt es in einem Zyklus für jeden Lautsprecher $i \in \{1, 2, 3, 4, 5\}$ jeweils nur einen Zeitschritt, in dem das Detonationssignal beginnt. Folglich gibt es auch nur einen Wert $u_{l,i} = 1$ pro Zyklus. Analog zur Ansteuerung der Lautsprecher können die Messungen an den zehn Messstellen im Zeitschritt l eines Zyklus im Vektor

$$\boldsymbol{\gamma}_l = [y_{l,1}, \dots, y_{l,j}, \dots, y_{l,10}]^T \quad (6.102)$$

zusammengefasst werden. Mit der Abtastfrequenz $f_u = f_y = 10$ kHz für Aktuation und Messung und einer angenommenen Feuerfrequenz von $f_f = 20$ Hz ergeben sich $n_u = n_y = n = 500$ Zeitschritte in einem Zyklus für jeden der fünf Lautsprecher und jedes der zehn Mikrofone. Die gesamte Aktuation aller Lautsprecher sowie die Messungen aller Mikrofone können in einem Zyklus durch die Supervektoren

$$\boldsymbol{\nu} = \begin{bmatrix} \boldsymbol{\nu}_1 \\ \vdots \\ \boldsymbol{\nu}_{n_u} \end{bmatrix} = \begin{bmatrix} \nu_1 \\ \nu_2 \\ \vdots \\ \nu_{2500} \end{bmatrix} \in \mathbb{B}^{500 \cdot 5} \quad \text{und} \quad \boldsymbol{\gamma} = \begin{bmatrix} \boldsymbol{\gamma}_1 \\ \vdots \\ \boldsymbol{\gamma}_{n_y} \end{bmatrix} \in \mathbb{R}^{500 \cdot 10} \quad (6.103)$$

repräsentiert werden.

Um ein Modell zu bestimmen, welches die Ansteuerung $\boldsymbol{\nu}$ auf den Systemausgang $\boldsymbol{\gamma}$ abbildet, wird zunächst jeder Lautsprecher zyklisch mit einem Einheitsimpuls im Zeitschritt l angeregt bis sich ein periodisches Verhalten in den Messungen zeigt. Bei Anregung des i -ten Lautsprechers ergibt sich dann die periodische Messung \mathbf{I}_i^l in Supervektordarstellung. Mit den gemessenen Impulsantwort \mathbf{I}_i^l lässt sich ein Modell:

$$\boldsymbol{\gamma} = \underbrace{[\mathbf{I}_1^1, \mathbf{I}_2^1, \dots, \mathbf{I}_5^1, \mathbf{I}_1^2, \mathbf{I}_2^2, \dots, \mathbf{I}_5^2, \dots, \mathbf{I}_1^n, \mathbf{I}_2^n, \dots, \mathbf{I}_5^n]}_{\boldsymbol{\Psi}} \boldsymbol{\nu} \quad (6.104)$$

aufbauen. Es ist anzumerken, dass nicht für jeden Zeitschritt l das Experiment mit einem Einheitsimpuls durchgeführt werden muss. Wenn alle Lautsprecher einmal angeregt wurden, können die anderen Impulsantworten durch eine einfache Verschiebung ermittelt werden (Wolff, 2018).

Wie in der Einleitung beschrieben, sollen mit der Regelung möglichst gleichmäßige Druckbedingungen im Plenum sichergestellt werden. Um die Druckschwankungen zu reduzieren, kann beispielsweise die bezüglich \mathbf{W}_E gewichteten 2-Norm verwendet werden. Wird von einer perfekten Modellierung und einem störungsfreien System ausgegangen, kann die optimale Ansteuerung der Lautsprecher $\boldsymbol{\nu}_\infty$ vor der eigentlichen Anwendung am Experiment bestimmt werden.

Quadratisches Programm (QP) 7: *Die optimale Steuerung des akustischen Systems ergibt sich aus*

$$\boldsymbol{\nu}_\infty = \arg \min_{\tilde{\boldsymbol{\nu}}} \tilde{\boldsymbol{\nu}}_\infty^T \boldsymbol{\Psi}^T \mathbf{W}_E \boldsymbol{\Psi} \tilde{\boldsymbol{\nu}}_\infty \quad (6.105)$$

unter der Beschränkung

$$\tilde{\boldsymbol{\nu}} = [\nu_1, \nu_1, \dots, \nu_{2500}]^T \in \mathbb{B}^{500 \cdot 5} \quad (6.106)$$

$$\|\mathbf{u}_i\| = 1 \quad \forall i \in \{1, 2, \dots, n_L\} \quad (6.107)$$

mit

$$\mathbf{u}_i = [\nu_i, \nu_{i+5}, \dots]^T \quad (6.108)$$

Die Beschränkung Gl. (6.107) stellt sicher, dass jeder Lautsprecher in einem Zyklus nur einmal einen Detonationsvorgang initiiert. In Wolff, 2018 werden die Möglichkeiten, mit einer optimalen Steuerung verschiedene Ziele für dieses System zu erreichen, ausführlich diskutiert. Für eine Anwendung mit Modellunsicherheiten und Störungen ist jedoch eine binäre ILR notwendig, um diese Einflüsse zu kompensieren.

Quadratisches Programm (QP) 8: *Die optimale Stellfolge $\boldsymbol{\nu}_{k+1}$ einer ILR für das akustische System ergibt sich aus*

$$\boldsymbol{\nu}_{k+1} = \arg \min_{\tilde{\boldsymbol{\nu}}_{k+1}} \tilde{\boldsymbol{\nu}}_{k+1}^T \mathbf{Q} \tilde{\boldsymbol{\nu}}_{k+1} + \mathbf{q}^T \tilde{\boldsymbol{\nu}}_{k+1} \quad (6.109)$$

unter der Beschränkung

$$\tilde{\boldsymbol{\nu}} = [\nu_1, \nu_1, \dots, \nu_{2500}]^T \in \mathbb{B}^{500 \cdot 5} \quad (6.110)$$

$$\|\mathbf{u}_i\| = 1 \quad \forall i \in \{1, 2, \dots, n_L\} \quad (6.111)$$

mit

$$\mathbf{u}_i = [\nu_i, \nu_{i+5}, \dots]^T \quad (6.112)$$

$$\mathbf{Q} = \Psi^T \mathbf{W}_E \Psi + \mathbf{W}_{\Delta U} \quad (6.113)$$

$$\mathbf{q}^T = -2 \left(\nu_k^T \left(\Psi^T \mathbf{W}_E \Psi + \mathbf{W}_{\Delta U} \right) - \gamma_k \mathbf{W}_E \Psi \right) \quad (6.114)$$

Damit muss sowohl für die Optimalsteuerung als auch in jeder Iteration der ILR ein binäres Optimierungsproblem der Dimension $n_L \cdot n_u = 2500$ gelöst werden.

6.5.2 Reduzierung des Rechenaufwands

Selbst mit den aktuell leistungsfähigsten Lösern wie *Gurobi* (Gurobi Optimization, 2020) oder *CPLEX* (Blick et al., 2014) erfordert eine Berechnung von QP 7 bzw. QP 8 eine Rechenzeit von mehreren Stunden auf einem aktuellen PC. Für eine optimale Steuerung gemäß QP 7 ist der Rechenaufwand bzw. der damit verbundene Zeitaufwand unkritisch. Für eine ILR wird jedoch für das betrachtete akustische System eine Lösung im Millisekundenbereich benötigt. Daher wird an dieser Stelle ein Ansatz vorgestellt, der den Rechenaufwand für dieses System deutlich reduzieren kann. Das grundlegende Konzept dieses Verfahrens wurde in Arnold, Topalovic und King, 2020 veröffentlicht. Dabei wird ausgenutzt, dass die Stellgröße ν_k in jedem Zyklus nur fünf Einsen enthält und ansonsten ausschließlich Nullen.

Da der folgende Ansatz für jede binäre ILR mit einer Stellfolge \mathbf{u} und einer Modellmatrix \mathbf{G} verwendet werden kann, wird hier zunächst die entsprechende, allgemeine Notation aus den vorherigen Abschnitten verwendet. Für den Fall des akustischen Plenumsversuchs gilt entsprechend $\mathbf{u} = \nu$ und $\mathbf{G} = \Psi$.

Ausgangspunkt für eine Reduzierung des Rechenaufwand ist wie bereits in Abschnitt 6.3.2 zur OMS eine Verkleinerung des zulässigen Lösungsraums auf eine Menge

$$\mathcal{S}_{k+1} = \{\mathbf{u}_{k+1}^1, \dots, \mathbf{u}_{k+1}^i, \dots, \mathbf{u}_{k+1}^{n_S}\} \quad (6.115)$$

welche die zulässigen Lösungen für den nächsten Zyklus $k+1$ enthält. Dabei wurde abermals angenommen, dass die Lösungsmenge aus einer zyklusinvarianten Anzahl $n_S = |\mathcal{S}_{k+1}| \forall k \in \{0, 1, \dots\}$ an potentiellen Lösungen

$$\mathbf{u}_{k+1}^i = [u_{k+1,1}^i, \dots, u_{k+1,i}^i, \dots, u_{k+1,n_u}^i]^T \quad (6.116)$$

besteht. Ausgehend von allen n_S potentiellen Lösungen kann ein Vektor

$$\mathbf{p}_{k+1} = [p_{k+1,1}, \dots, p_{k+1,i}, \dots, p_{k+1,n_u}]^T \quad (6.117)$$

aufgebaut werden, wobei für die einzelnen Elemente

$$p_{k+1,i} = \begin{cases} 1 & \text{falls } \exists \mathbf{u}_{k+1}^i \in \mathcal{S}_{k+1} \mid u_{k+1,i}^i = 1 \\ 0 & \text{sonst} \end{cases} \quad (6.118)$$

gilt. Der Vektor \mathbf{p}_{k+1} hat folglich Einsen an den Stellen, für die eine Lösung in \mathcal{S}_{k+1} existiert, die an dieser Stelle eine Eins hat. Aus dem Vektor \mathbf{p}_{k+1} kann eine entsprechende Diagonalmatrix

$$\mathbf{D}_{k+1} = \text{diag}(\mathbf{p}_{k+1}) = [\mathbf{d}_1, \dots, \mathbf{d}_i, \dots, \mathbf{d}_{n_u}] \quad (6.119)$$

mit den Spalten \mathbf{d}_i aufgebaut werden. Werden die Spalten $\mathbf{d}_i = \mathbf{0}$, die komplett aus Nullen bestehen, aus der Matrix \mathbf{D}_{k+1} entfernt, ergibt sich eine transponierte Transformationsmatrix \mathbf{T}_{k+1}^T , mit der die tatsächliche Stellfolge \mathbf{u}_{k+1} auf eine Ersatzstellfolge

$$\bar{\mathbf{u}}_{k+1} = \mathbf{T}_{k+1} \mathbf{u}_{k+1} \quad (6.120)$$

transformiert werden kann. Durch die spezielle Struktur der Transformationsmatrix \mathbf{T}_{k+1} gilt zusätzlich

$$\mathbf{T}_{k+1}^T \bar{\mathbf{u}}_{k+1} = \mathbf{u}_{k+1} \quad . \quad (6.121)$$

Damit ist neben Transformation auch eine Rücktransformation definiert.

Aus der Zusammensetzung des Vektors \mathbf{p}_{k+1} ergibt sich, dass die Dimension der Ersatzstellfolge $\bar{\mathbf{u}}_{k+1}$ der Anzahl $n_1(\mathbf{p}_{k+1})$ an Einsen im Vektor \mathbf{p}_{k+1} entspricht. Folglich reduziert sich die Dimension von $\bar{\mathbf{u}}_{k+1}$ für den Fall $n_1(\mathbf{p}_{k+1}) \ll n_u$ signifikant. Durch die Transformation ergibt sich ein angepasstes Optimierungsproblem.

Quadratisches Programm (QP) 9: Die optimale Ersatzstellfolge $\bar{\mathbf{u}}_{k+1}$ ergibt sich aus

$$\bar{\mathbf{u}}_{k+1} = \arg \min_{\tilde{\bar{\mathbf{u}}}_{k+1}} \tilde{\bar{\mathbf{u}}}_{k+1}^T \bar{\mathbf{Q}} \tilde{\bar{\mathbf{u}}}_{k+1} + \bar{\mathbf{q}}^T \tilde{\bar{\mathbf{u}}}_{k+1} \quad (6.122)$$

unter der Bedingung

$$\tilde{\bar{\mathbf{u}}}_{k+1} \in \mathcal{U}_T \subsetneq \mathbb{B}^{n_1(\mathbf{p}_{k+1})} \quad (6.123)$$

mit

$$\bar{\mathbf{Q}} = \mathbf{T}_{k+1} \left(\mathbf{G}^T \mathbf{W}_E \mathbf{G} + \mathbf{W}_{\Delta U} \right) \mathbf{T}_{k+1}^T \quad (6.124)$$

$$\bar{\mathbf{q}}^T = -2 \left(\mathbf{u}_k^T \mathbf{G}^T \mathbf{W}_E \mathbf{G} + \mu_k^T \mathbf{W}_{\Delta U} + \mathbf{e}_k^T \mathbf{W}_E \mathbf{G} \right) \mathbf{T}_{k+1}^T \quad . \quad (6.125)$$

Die Menge \mathcal{U}_T muss entsprechend der Transformation bestimmt werden, falls zusätzliche Beschränkungen gelten.

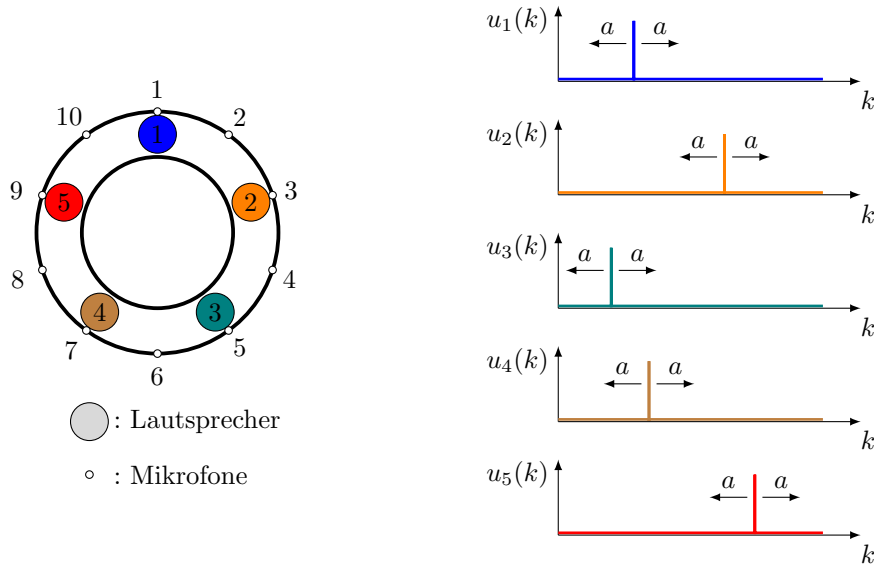
Im Fall des akustischen Plenumsversuchs muss der Unterraum \mathcal{U}_T entsprechend der Beschränkung aus Gl. (6.107) bestimmt werden. Diese Beschränkung kann als lineare Matrixungleichung ausgedrückt werden (Wolff, 2018). Für lineare Beschränkungen, die durch eine Matrix $\mathbf{\Omega}$ und einen Vektor \mathbf{c} definiert sind, ergibt sich die Umformung

$$\mathbf{\Omega} \mathbf{u}_{k+1} \leq \mathbf{c} \quad (6.126)$$

$$\mathbf{\Omega} \mathbf{T}_{k+1}^T \bar{\mathbf{u}}_{k+1} \leq \mathbf{c} \quad , \quad (6.127)$$

die dann einen entsprechenden transformierten Lösungsraum \mathcal{U}_T definiert. Je nach Wahl der Lösungsmenge \mathcal{S}_{k+1} ist es möglich, dass bestimmte Beschränkungen komplett wegfallen.

Die Transformation und die damit verbundene Reduzierung der Dimension des Optimierungsproblems beeinflusst den Rechenaufwand über zwei verschiedene Aspekte. Zum einen kann die relaxierte Lösung in jedem Explorationsschritt des ganzzahligen QPs schneller bestimmt werden, da das QP kleiner ist und möglicherweise Beschränkungen in der Transformation wegfallen. Zum anderen führt die reduzierte Dimension dazu, dass die maximale Anzahl der Lösungen, die in der *Branch-and-Bound*-Suche exploriert werden müssen, verringert wird. Wie stark der Rechenaufwand mit dieser Methode tatsächlich reduziert wird, hängt neben der Größe der Lösungsmenge \mathcal{S}_{k+1} vor allem, wie oben beschrieben, von dem Verhältnis aus Einsen und Nullen in den Lösungen aus \mathcal{S}_{k+1} ab.


 Abbildung 6.16: Zulässige Anpassung im Zyklus k in Abhängigkeit des Parameters a

6.5.3 Ergebnisse am akustischen Plenumsversuch

Um den beschriebenen Ansatz der reduzierten binären ILR am akustischen Plenumsversuch testen zu können, muss zunächst festgelegt werden, wie die Lösungsmenge \mathcal{S}_{k+1} im entsprechenden Zyklus zusammengestellt wird. Es wird davon ausgegangen, dass im Anschluss an die Modellidentifikation eine optimale Steuerung ν_∞ gemäß QP 7 bestimmt wird, die z.B. als initiale Stellfolge ν_0 verwendet werden kann. Da zu erwarten ist, dass die ILR im Betrieb nur leichte Störungen ausgleichen muss, kann angenommen werden, dass sich die Stellfolge über die Zyklen in der Nähe der optimalen Steuerung befindet. Ausgehend von dieser Annahme kann festgelegt werden, dass in jeder Iteration k die Zündzeit eines Rohres bzw. Ansteuerung des Lautsprechers nur in einem beschränkten Zeitfenster angepasst werden muss. Durch die Diskretisierung der Zeit und die Beschränkung aus Gl. (6.107) können die Zündungen der einzelnen Rohre in einem Zyklus k durch die Zeitschritte

$$l_k^1, l_k^2, l_k^3, l_k^4, l_k^5 \in \{1, 2, \dots, 500\} \quad (6.128)$$

repräsentiert werden. Soll die Veränderung der Zündzeiten nun wie beschrieben beschränkt werden, kann ein Parameter $a \in \mathbb{N}$ eingeführt werden, der angibt, um wieviele Zeitschritte die Zündzeit eines einzelnen Rohres im folgenden Zyklus $k+1$ angepasst werden darf. Für die Zündzeiten im nächsten Zyklus $k+1$ gilt dann

$$\begin{aligned} l_k^1 - a &\leq l_{k+1}^1 \leq l_k^1 + a \\ &\vdots \\ l_k^5 - a &\leq l_{k+1}^5 \leq l_k^5 + a \end{aligned} \quad (6.129)$$

Sollte sich bei der Anpassung ein Zeitschritt l_{k+1} ergeben, der nicht mehr innerhalb der Zeitschritte eines Zykluses liegt, muss der zyklische Charakter des Prozesses berücksichtigt werden. Ergibt sich ein $l_k + a > 500$ bzw. $l_k - a < 1$ kann der Wert auf den entsprechenden Wert am Anfang bzw. am Ende übertragen werden. Die Anpassungsregel in Abhängigkeit des Parameters a ist in Abbildung 6.16 illustriert. Da im Supervektor ν_k die Zeitschritte für die Zündungen in binärer Form repräsentiert sind, ist es möglich, auch die potentiellen binären

Stellfolgen ν_{k+1}^i zu definieren, die gemäß Gl. (6.129) zulässige Lösungen sind. Alle zulässigen Lösungen ν_{k+1}^i können in der Menge \mathcal{S}_{k+1} zusammengefasst und für die Reduzierung entsprechend Abschnitt 6.5.2 verwendet werden. Dabei ist wichtig hervorzuheben, dass es nicht notwendig ist, alle zulässigen Kombinationen aus den Zündzeiten der einzelnen Rohre in der Lösungsmenge zu berücksichtigen, die für die Reduzierung benutzt wird. Gemäß Gl. (6.118) wird in den Transformationsvektor \mathbf{p}_{k+1} eine Eins eingetragen, sobald ein einziger Vektor in \mathcal{S}_{k+1} existiert, der an der entsprechenden Stelle ebenfalls eine Eins aufweist. Daher müssen nicht alle Kombinationen berücksichtigt werden. Als Resultat hängt die Kardinalität der Lösungsmenge lediglich linear vom Parameter a ab:

$$|\mathcal{S}_{k+1}| = n_L \cdot (2a + 1) = 5 \cdot (2a + 1) \quad (6.130)$$

Der Parameter a beeinflusst damit direkt den Rechenaufwand eines einzelnen Schritts der ILR.

Wie bereits beschrieben, ist das Ziel der ILR, die Druckschwankungen zu minimieren. Dazu wird wie in QP 7 und QP 8 zu sehen, der Systemausgang direkt als Regelfehler angenommen $\mathbf{e}_k = -\gamma$. Mit dieser Referenz führt die Anwendung der ILR mit der beschriebenen Kostenfunktion zu einer Reduzierung der mit \mathbf{W}_E gewichteten 2-Norm aller gemessenen Mikrofon-signale.² Für einen ersten Versuch wurde der Parameter $a = 2$ gewählt. Damit reduziert sich die Dimension des Optimierungsproblems von $n_L \cdot n_u = 2500$ auf $n_L \cdot \bar{n}_u = 25$. Durch diese Reduzierung um zwei Größenordnungen ist das binäre Optimierungsproblem nun mit deutlich kleinerem Rechenaufwand über eine *Branch-and-Bound*-Suche zu lösen.

Um zu untersuchen, wie sich der Regler für verschiedene Abweichungen von der optimalen Stellfolge verhält, werden verschiedene Initialwerte verwendet. Abbildung 6.17 zeigt den mittleren quadratischen Fehler aller zehn Mikrofonmessungen über die Zyklen, ausgehend von einer simultanen Feuerung aller Rohre bzw. eines zeitlich äquidistant verteilten Feuermusters. Zusätzlich wurde ein initiales Muster gewählt, welches nur leicht von der global optimalen Steuerung abweicht. Die Ergebnisse zeigen einen deutlichen Unterschied in Ab-

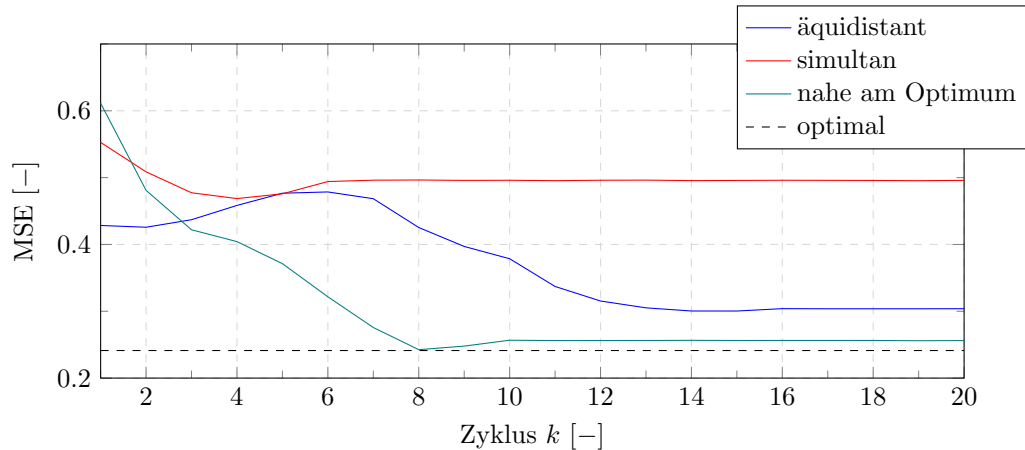


Abbildung 6.17: Mittlerer quadratischer Fehler (*Mean Squared Error*, MSE) über alle 10 Messpositionen. Als initiale Stellfolge wurden zwei generische Feuermuster (zeitlich äquidistant und simultan) sowie eine leichte Perturbation der optimalen Steuerung genutzt.

hängigkeit von der initialen Stellfolge. Besonders auffällig ist, dass in allen drei Fällen kein

²Bei passender Wahl der Gewichtung \mathbf{W}_E kann auch eine Minimierung der Varianz erreicht werden, siehe Wolff, 2018

monotones Konvergenzverhalten zu beobachten ist. Dies ist darauf zurückzuführen, dass der Ansatz der Modelltransformation nicht die Informationen über das tatsächliche Optimum erhalten kann. Die Betrachtung des Optimierungsproblems in einem reduzierten Unterraum kann durch den Verlust an Informationen keine optimale Lösung garantieren. Folglich wird das Optimum des reduzierten Problems nicht zwangsläufig auf das Optimum des vollen Problems zurücktransformiert, selbst wenn die optimale Lösung des vollen Problems in der Lösungsmenge enthalten ist. Dies ist besonders deutlich im Verlauf der Stellfolge zu sehen, die nahe an der optimalen Steuerung startet. In Zyklus $k = 8$ wird als Lösung des reduzierten Problems die optimale Lösung des vollen Problems gefunden. In Zyklus $k = 9$ liefert das neu angepasste reduzierte Problem wiederum eine andere Stellfolge als optimal. Obwohl die Optimalität nicht gewährleistet ist, kann mit der ILR trotzdem in allen Fällen der Fehler, also die Druckschwankung, deutlich reduziert werden. Die abweichenden Ergebnisse zeigen jedoch die starke Abhängigkeit von der initialen Stellfolge. Die verschiedenen Feuermuster, die sich im konvergierten Zustand ergeben, sind in Abb. 6.18 gezeigt. Auch hier zeigt sich der

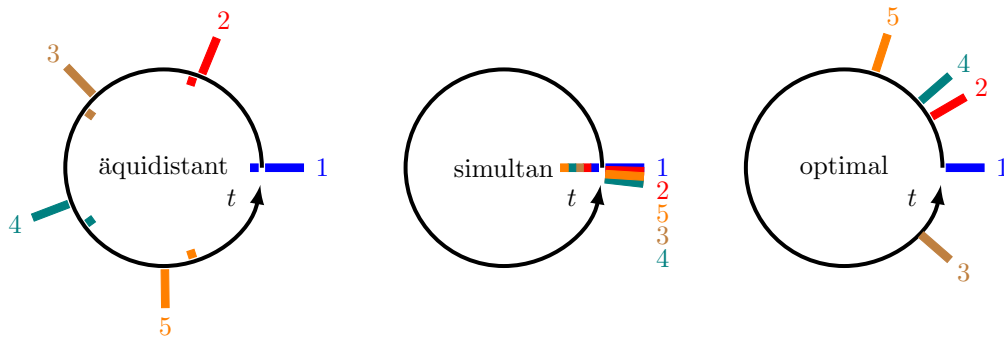


Abbildung 6.18: Konvergierte Feuermuster, die sich für verschiedene initiale Feuermuster (zeitlich äquidistant und simultan) mit einer reduzierten binären ILR ergeben. Die initialen Feuermuster sind innen gezeigt. Zusätzlich ist das optimale Feuermuster gezeigt.

deutliche Unterschied in den konvergierten Zuständen.

Um den Einfluss der Größe des Lösungsraums zu untersuchen, ist in Abb. 6.19 der mittlere quadratische Fehler für Parameter $a = 2$ und $a = 5$, jeweils ausgehend von einem simultanen bzw. zeitlich äquidistanten initialen Feuermuster, gezeigt. Der Verlauf für ein initiales Feu-

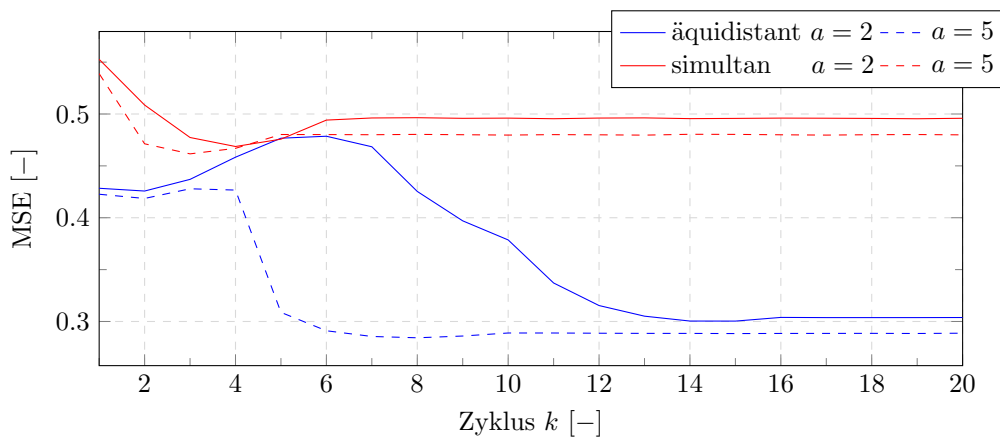


Abbildung 6.19: Mittlerer quadratischer Fehler (*Mean Squared Error*, MSE) über alle 10 Messpositionen. Als initiale Stellfolge wurden ein simultanes und zeitlich äquidistantes Feuermuster genutzt. Die Kardinalität des Lösungsraum variiert in Abhängigkeit des Parameters a .

ermuster in der Nähe der optimalen Steuerung ist hier nicht gezeigt, da sich für diesen Fall im Experiment kein starker Einfluss des Parameters a gezeigt hat. Für die anderen beiden initialen Stellfolgen ergeben sich jedoch jeweils leicht verbesserte, konvergierte Ergebnisse. Besonders auffällig ist, das stark veränderte Konvergenzverhalten im Fall eines zeitlich äquidistanten, initialen Feuermusters. Hier konnte durch eine Vergrößerung des Lösungsraums eine deutliche Beschleunigung des Konvergenzverhaltens erzielt werden. Die weniger starke Reduzierung des Optimierungsproblems führt dazu, dass mehr Informationen aus dem realen Optimierungsproblem erhalten bleiben. Der Nebeneffekt dieses verbesserten Ergebnisses ist jedoch ein deutlich erhöhter Rechenaufwand. Gemäß Gl. (6.130) wächst die Kardinalität des Lösungsraums linear-affin auf $|\mathcal{S}| = 55$ an. Weitere Experimente haben gezeigt, dass eine weitere Vergrößerung des Lösungsraums die Ergebnisse zwar nicht mehr so stark, jedoch in allen Fällen weiter verbessert. Allerdings ist für größere Lösungsmengen eine Berechnung in Echtzeit nicht mehr möglich. Für die Wahl des Lösungsraums ist immer eine Abwägung zwischen Rechenaufwand und der Güte der Regelung notwendig. Der Lösungsraum sollte immer so gewählt werden, dass die verfügbare Rechenleistung voll ausgeschöpft wird, um ein möglichst gutes Ergebnis zu erzielen.

Eine der für die Anwendung besonders praktischen Eigenschaften einer normoptimalen ILR ist die Möglichkeit, auch Linearisierungsfehler in der Modellierung über die Iterationen zu kompensieren. Voraussetzung dafür ist ein zyklusinvarianter Linearisierungsfehler, sodass die entstehende Abweichung mit im Versatz \mathbf{b} des angenommenen linear-affinen Modells

$$\mathbf{y}_k = \mathbf{G} \cdot \mathbf{u}_k + \mathbf{b} \quad (6.131)$$

berücksichtigt werden kann. Wie in Abschnitt 6.2 zur Theorie der ILR beschrieben, wird die Differenz aus Referenz und Offset über die Zyklen ausgelernt. Damit kann der lineare ILR-Ansatz mitunter auch für nichtlineare Systeme verwendet werden. Dies ist besonders relevant für eine mögliche Anwendung der ILR in einem Plenumsaufbau mit tatsächlichen Detonationsrohren. Im Gegensatz zu dem verwendeten Ersatzversuchstand mit annähernd linearen, akustischen Eigenschaften ist zu erwarten, dass in der strömungsmechanischen Anwendung mit Detonationsrohren nicht mehr von einer linearen Systembeschreibung ausgegangen werden kann.

Um eine Einschätzung zu erhalten, ob der Ansatz der reduzierten ILR auch für nichtlineare Systeme eine Kompensation der Druckschwankungen ermöglicht, wird dem akustischen System eine künstliche Ausgangsnichtlinearität hinzugefügt. Der künstliche, nichtlineare Ausgang $y_n(t)$ ergibt sich aus

$$y_n(t) = 5 \cdot \text{sgn}(y(t)) \cdot y(t)^2 \quad . \quad (6.132)$$

Hier bezeichnet $\text{sgn}(\cdot)$ die Signum-Funktion, die notwendig ist, um die Vorzeicheninformation zu erhalten. Andernfalls wäre ein Ausgleich der Druckspitzen gegen Drucksinken nicht möglich. Abbildung 6.20 zeigt die Ergebnisse mit der künstlichen Ausgangsnichtlinearität ausgehend von einem zeitlich äquidistanten, initialen Feuermuster und einem initialen Feuermuster in der Nähe des linearen Optimums. Für das initiale Feuermuster in der Nähe der optimalen Lösung ergibt sich nur eine minimale Verbesserung in den ersten zwei Schritten. Offensichtlich ist hier eine signifikante Verbesserung im Bereich des gewählten Lösungsraums nicht möglich. Im Gegensatz dazu lässt sich für den Fehlerverlauf ausgehend von einem zeitlich äquidistanten Feuermuster eine deutliche Verbesserung erkennen. Die Ergebnisse zeigen qualitativ ein ähnliches Verhalten wie im linearen Fall. Die geringere Größenordnung des Fehlers ergibt sich aus der vorgeschalteten Nichtlinearität. Es ergibt sich zwar auch hier kein monotonen Konvergenzverhalten, allerdings konvergiert der Fehler auch für das angenommene

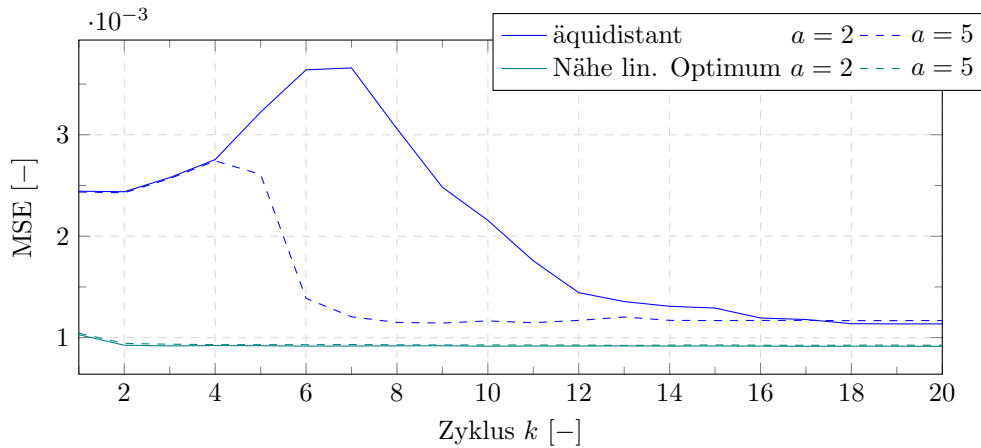


Abbildung 6.20: Mittlerer quadratischer Fehler (*Mean Squared Error*, MSE) über alle 10 Messpositionen. Es wurde eine künstliche Ausgangsnichtlinearität angenommen. Als initiale Stellfolge wurde ein zeitlich äquidistantes Feuermuster und ein Feuermuster in der Nähe der linear optimalen Lösung genutzt. Die Kardinalität des Lösungsraum variiert in Abhängigkeit des Parameters a .

nichtlineare System. Interessanterweise ergibt sich in diesem Fall sogar ein kleinerer konvergierter Regelfehler mit einem kleineren Lösungsraum. Dies hebt noch einmal den schwierig vorherzusagenden Einfluss der Wahl des Lösungsraums hervor.

6.5.4 Wahl der Lösungsmenge

Im vorangegangenen Abschnitt wurde immer von einer sehr einfachen Definition der Lösungsmenge \mathcal{S}_{k+1} in Abhängigkeit des Parameters a ausgegangen. Dabei war lediglich das Ziel, eine Zusammenstellung der Lösungsmenge zu finden, welche die Zündzeiten zwischen zwei Zyklen nur in einem beschränkten Maß anpasst, um damit Echtzeitfähigkeit zu garantieren. Auch wenn dieser Ansatz bereits zufriedenstellende Ergebnisse liefert, ist jedoch die Konvergenzgeschwindigkeit stark beschränkt. Im Folgenden soll genauer auf die Wahl der Lösungsmenge \mathcal{S}_{k+1} eingegangen und ein Weg aufgezeigt werden, mit dem ein deutlich schnelleres Konvergenzverhalten erzielt werden kann. Die Umsetzung dieses Konzeptes wurde in Topalovic, Arnold und King, 2022 veröffentlicht.

In der Praxis hat sich gezeigt, dass es stark von dem zu regelnden System abhängt, welche Zusammenstellung der Lösungsmenge \mathcal{S}_{k+1} zu einem besonders guten Regelverhalten führt. Generell kann jedoch festgehalten werden, dass eine größere Lösungsmenge \mathcal{S}_{k+1} zumeist zu einer schnelleren Konvergenz des Reglers führt. Mit einer wachsenden Lösungsmenge \mathcal{S}_{k+1} sinkt allerdings auch der Vorteil in der Rechenzeit, der durch die Reduzierung erzielt wird. Eine sichere Konvergenz zur optimalen Lösung kann in allen Fällen aber nur für die vollständige ILR erwartet werden.

Ziel bei der Zusammenstellung der Lösungsmenge \mathcal{S}_{k+1} muss es sein, potentielle Stellfolgen auszuwählen, von denen eine besonders große und zuverlässige Verbesserung bezüglich der Kostenfunktion zu erwarten ist. Dies kann z.B. über die Betrachtung der Sensitivität der Kostenfunktion erfolgen, mit dem Ziel die Hauptrichtungen zu bestimmen, in die eine besonders starke Veränderung zu erwarten ist. Dieser Ansatz hat sich in diversen Untersuchungen im Rahmen dieser Arbeit als besonders zielführend erwiesen.

Ausgangspunkt ist die Bestimmung der Eigenvektoren \mathbf{v}_i und Eigenwerte λ_i des quadratischen

Parameters \mathbf{Q} . Dafür wird zunächst die Singulärwertzerlegung berechnet:

$$\mathbf{Q} = \mathbf{V} \mathbf{S} \mathbf{V}^T \quad (6.133)$$

Diese spezielle Form der Singulärwertzerlegung gilt nur, da die Matrix \mathbf{Q} positiv definit und symmetrisch ist. Folglich entsprechen Eigenwerte³ bzw. Eigenvektoren den Singulärwerten bzw. Spalten der Matrix \mathbf{V} . Die Eigenvektoren \mathbf{v}_i als Spalten der Matrix

$$\mathbf{V} = [\mathbf{v}_1, \dots, \mathbf{v}_i, \dots, \mathbf{v}_{n_u}] \quad (6.134)$$

definieren die Verdrehung des Paraboloids, also der geometrischen Repräsentation der Kostenfunktion, gegenüber einem kartesischen Koordinatensystem. Im Folgenden wird davon ausgegangen, dass die Eigenvektoren absteigend sortiert sind entsprechend der Größe des zugehörigen Eigenwerts. Zusätzlich wird angenommen, dass die Eigenvektoren auf $\|\mathbf{v}_i\|_2 = 1$ normiert sind. Das Produkt $\mathbf{v}_i \cdot \lambda_i$ wird als skaliertter Eigenvektor \mathbf{v}_i^s bezeichnet. Abbildung 6.21 illustriert diese Transformation exemplarisch für eine zweidimensionale Stellfolge. Zusätzlich definieren die Eigenwerte, wie ebenfalls in Abb. 6.21 dargestellt, die Verzerrung des Paraboloids. Damit bestimmen die Matrizen \mathbf{V} und \mathbf{S} die Form des Paraboloids. Die Eigenvektoren,

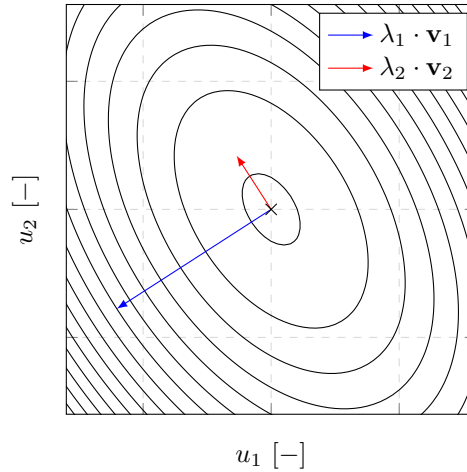


Abbildung 6.21: Geometrischer Zusammenhang zwischen dem quadratischen Parameter \mathbf{Q} und seinen Eigenvektoren bzw. Eigenwerten.

die zu den größten Eigenwerten gehören, geben die Richtung des steilsten Abstiegs an. Entsprechend sind dies die Richtungen, in die ohne weitere Informationen eine besonders starke Veränderung in der Kostenfunktion zu erwarten ist. Folglich ist z.B. von einer Anpassung der aktuellen Stellgröße

$$\mathbf{u}_{k+1} = \mathbf{u}_k \pm \mathbf{v}_1 \quad (6.135)$$

zu erwarten, dass sich die Kostenfunktion besonders stark ändert. Ungünstigerweise ist eine Anpassung wie in Gl. (6.135) im ganzzahligen bzw. binären Fall nicht möglich, da der Eigenvektor \mathbf{v}_1 reellwertig ist und sich somit eine Stellfolge \mathbf{u}_{k+1} ergibt, die bezüglich der Beschränkung nicht zulässig ist.

Um trotzdem eine möglichst starke Anpassung in Richtung der größten Hauptkomponenten zu ermöglichen, kann eine Kennzahl

$$d = f_d(\mathbf{v}_i, \Delta \mathbf{u}_{k+1}) \quad (6.136)$$

³Da \mathbf{Q} positiv definit ist, sind alle Eigenwerte positiv.

eingeführt werden, welche die Ähnlichkeit zwischen einem Eigenvektor \mathbf{v}_i und einer zulässigen Änderung der Stellfolge $\Delta \mathbf{u}_{k+1}$ mit einem skalaren Wert d bewertet. Diese Kennzahl kann dann genutzt werden, um z.B. möglichst ähnliche Vektoren zu den größten skalierten Eigenvektoren zu finden, um daraus die Lösungsmenge \mathcal{S}_{k+1} zusammen zu stellen. Wie genau die Ähnlichkeit zwischen zwei Vektoren bemessen wird, muss abhängig von der konkreten Problemstellung entschieden werden. Soll ausschließlich die Richtung der beiden Vektoren möglichst ähnlich sein, bietet sich z.B. die Kosinus-Ähnlichkeit an. Gelten jedoch weitere Beschränkungen für \mathbf{u}_{k+1} und damit auch für $\Delta \mathbf{u}_{k+1}$ kann die Definition einer passenden Kennzahl deutlich erschwert werden.

Ein weiterer Aspekt bei der Definition der Kennzahl ist der Rechenaufwand, der notwendig ist, um in jedem Zyklus die Lösungsmenge \mathcal{S}_{k+1} mithilfe der Funktion aus Gl. (6.136) zu bestimmen. Muss zur Bestimmung der Menge die Funktion aus Gl. (6.136) für jede Lösung des vollständigen Problems ausgewertet werden, ergibt sich ein Rechenaufwand, der die Vorteile der Reduzierung der Dimension des Optimierungsproblems negiert. Dies wäre z.B. bei dem Vergleich über die Kosinus-Ähnlichkeit zwischen den Vektoren der Fall.

Um aufzuzeigen, wie eine Kennzahl für ein konkretes Problem bestimmt werden kann, wird im Folgenden nochmal auf den akustischen Plenumsversuch eingegangen. Für das Beispiel wird ein Ähnlichkeitsmaß eingeführt, welches es erlaubt, effizient die Menge der Lösungen \mathcal{S}_{k+1} zusammenzustellen, ohne die Funktion aus Gl. (6.136) für alle Lösungen auszuwerten.

Beispiel akustischer Plenumsversuch

Für die Stellfolge am akustischen Plenumsversuch gilt die Beschränkung aus Gl. (6.107), welche sicherstellt, dass jedes Rohr nur einmal pro Zyklus gezündet wird. Dadurch ist der Raum der zulässigen Lösungen des vollständigen Problems bereits stark eingeschränkt. Umso schwerer ist es, auf einem einfachen Weg zulässige Änderungen $\Delta \mathbf{u}_{k+1}$ auszuwählen, die den größten skalierten Eigenvektoren möglichst ähnlich sind. Tatsächlich wird es im allgemeinen Fall gar nicht möglich sein, nur den größten, skalierten Eigenvektor zu betrachten. Vielmehr müssen alle skalierten Eigenvektoren herangezogen werden. Schließlich ergibt sich aus Gl. (6.107) eine Beschränkung für zulässige Lösungen $\Delta \mathbf{u}_{k+1}$. Zulässige Änderungen $\Delta \mathbf{u}_{k+1}$ haben einen Wert von minus eins an der Stelle, an der \mathbf{u}_k einen Wert von eins hat und einen Wert von eins an einer beliebigen, anderen Stelle. Dabei muss natürlich berücksichtigt werden, dass die Stelle in \mathbf{u}_k , an der die neue Eins auftaucht, auch zu dem entsprechenden Rohr gehört.

Für die Ähnlichkeitsbeziehung zu den Eigenvektoren, werden nun jeweils die minimalen und maximalen Werte in den skalierten Eigenvektoren und der zulässigen Änderung verglichen. Damit soll sichergestellt werden, dass der Wert minus eins in $\Delta \mathbf{u}_{k+1}$ an der Stelle des kleinsten Wertes im skalierten Eigenvektor liegt und entsprechend der Wert eins an der Stelle des größten Wertes im skalierten Eigenvektor liegt. Angenommen die Funktion $\sigma_{\text{Ind}}(\cdot)$ gibt den Index des maximalen Wertes eines Vektors zurück, dann ist das Ziel, die größten skalierten Eigenvektoren \mathbf{v}_i^s zu finden, für die

$$\sigma_{\text{Ind}}(\Delta \mathbf{u}_{k+1}) = \sigma_{\text{Ind}}(\mathbf{v}_i^s) \quad (6.137)$$

$$\sigma_{\text{Ind}}(-\Delta \mathbf{u}_{k+1}) = \sigma_{\text{Ind}}(-\mathbf{v}_i^s) \quad (6.138)$$

gilt.

Aus den entsprechenden zulässigen Änderungen $\Delta \mathbf{u}_{k+1}$ kann dann die Lösungsmenge \mathcal{S}_{k+1} für die Reduzierung des Optimierungsproblems bestimmt werden. Der Ablauf für die Erstellung

der Lösungsmenge \mathcal{S}_{k+1} ist in Algorithmus 2 gezeigt. Der Algorithmus bezieht sich für ein einfacheres Verständnis auf ein System mit lediglich einem Rohr kann aber problemlos unter der Berücksichtigung des Aufbaus von ν gemäß Gl. (6.103) auf das beschriebene System aus fünf Rohren erweitert werden. Auch in diesem Beispiel kann die Größe der Lösungsmenge über den Parameter a angepasst werden.

Algorithmus 2 Zusammenstellung der Lösungsmenge \mathcal{S}_{k+1} auf Basis der Eigenvektoren

```

1: Initialisierung  $\mathcal{S}_{k+1} = \{\mathbf{u}_k\}$ 
2:  $\{\mathbf{v}_1, \dots, \mathbf{v}_n\} := \text{Eigenvektoren}(\mathbf{Q})$ 
3:  $\{\lambda_1, \dots, \lambda_n\} := \text{Eigenwerte}(\mathbf{Q})$ 
4:  $\mathcal{P} := \{\mathbf{v}_1^s, \dots, \mathbf{v}_{2 \cdot n}^s\} = \{\lambda_1 \cdot \mathbf{v}_1, \dots, \lambda_n \cdot \mathbf{v}_n, -\lambda_1 \cdot \mathbf{v}_1, \dots, -\lambda_n \cdot \mathbf{v}_n\}$ 
5:  $z = 0$ 
6: while  $z < a$  do
7:   Initialisierung  $\Delta \mathbf{u} = \mathbf{0}$ 
8:    $l_- :=$  Zeitschritt, in dem das Rohr im Zyklus  $k$  gezündet wurde
9:    $\tilde{\mathbf{v}}^s :=$  Eigenvektor mit dem kleinsten Wert in Zeile  $l_-$ 
10:   $\Delta \mathbf{u}[l_-] = -1$ 
11:   $l_+ :=$  Zeilenindex des größten Werts in  $\tilde{\mathbf{v}}^s$ 
12:   $\Delta \mathbf{u}[l_+] = 1$ 
13:   $\tilde{\mathbf{u}}_{k+1} = \mathbf{u}_k + \Delta \mathbf{u}$ 
14:  if  $\tilde{\mathbf{u}}_{k+1} \notin \mathcal{S}_{k+1}$  then
15:     $\tilde{\mathbf{u}}_{k+1}$  zu  $\mathcal{S}_{k+1}$  hinzufügen
16:     $z = z + 1$ 
17:  end if
18:   $\tilde{\mathbf{v}}^s$  aus  $\mathcal{P}$  entfernen
19: end while

```

Wie stark diese Zusammenstellung der Lösungsmenge das Verhalten der reduzierten ILR beeinflusst, ist in Abb. 6.22 zu erkennen.⁴ Der dargestellte Verlauf des Regelfehlers über die

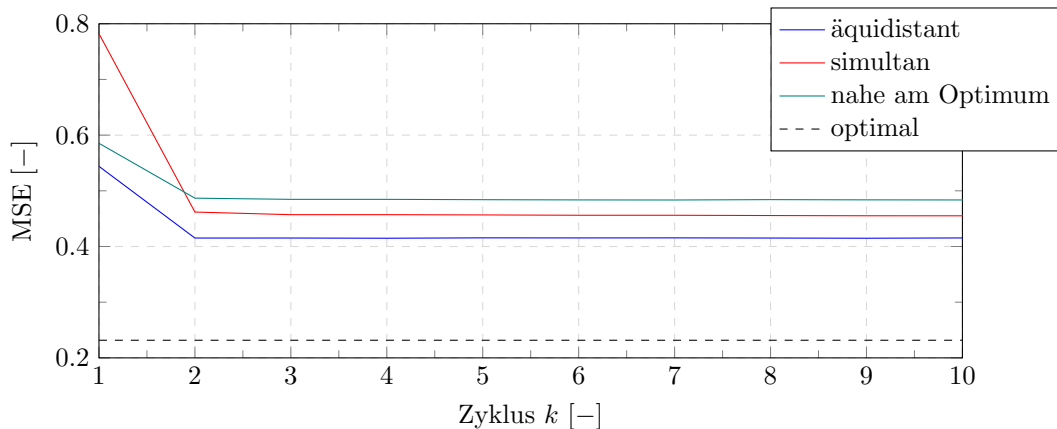


Abbildung 6.22: Mittlerer quadratischer Fehler (*Mean Squared Error*, MSE) über alle zehn Messpositionen. Es wurde eine reduzierte, binäre ILR mit einer Lösungsmenge gemäß Algorithmus 2 verwendet. Als initiale Stellfolge wurde ein zeitlich äquidistantes Feuermuster und ein Feuermuster in der Nähe der optimalen Lösung genutzt. Die Kardinalität des Lösungsraum ist über $a = 4$ definiert.

⁴Die abweichenden Startwerte gegenüber den Verläufen in Abb. 6.17 resultiert aus schwankenden Umgebungsbedingungen am Versuchsaufbau. Die Messungen wurden mit einem Abstand von fast einem Jahr durchgeführt.

Zyklen zeigt ein beschleunigtes Konvergenzverhalten im Vergleich zu den in Abschnitt 6.5.3 vorgestellten Ergebnissen. Allerdings kann für keinen der betrachteten Fälle das gleiche Niveau wie mit dem Lösungsraum aus Abschnitt 6.5.3 erreicht werden. Dies ist mit dem Fokus auf besonders starke und damit schnelle Anpassungen bei der Zusammenstellung der Lösungsmenge über Algorithmus 2 zu erklären. Zwar führt dieser Ansatz zu einer schnellen Korrektur durch die Regelung, eine feinere Anpassung, um die optimale Lösung zu erreichen, ist allerdings nicht möglich.

Das Beispiel zeigt, wie stark das Konvergenzverhalten des Ansatzes über die Wahl der Lösungsmenge beeinflusst werden kann. Da die Lösungsmenge nicht in jedem Zyklus mit der gleichen Strategie zusammengestellt werden muss, kann dieses Verhalten auch genutzt werden, um die Regelung in unterschiedlichen Situationen anzupassen. So könnte beispielsweise eine Lösungsmenge gemäß Algorithmus 2 genutzt werden, um schnell auf plötzliche Störungen zu reagieren. Wenn sich damit keine weitere Verbesserung ergibt, könnte zu einer Lösungsmenge, wie in Abschnitt 6.5.3 beschrieben, gewechselt werden, um eine feinere Einstellung in Richtung der optimalen Lösung zu ermöglichen. Auch eine Durchmischung der Lösungsmenge über unterschiedliche Verfahren in einem einzelnen Zyklus ist vorstellbar.

7 Extremwertregelung

Die in Kapitel 6 präsentierten Ergebnisse zeigen die besondere Leistungsfähigkeit der ILR. Allerdings hängen diese Ergebnisse stark von der Qualität des angenommenen Modells ab. In Fällen, in denen die Modellierung zu einem zu aufwendigen oder ungenauen Modell führt, sowie für Regelungsaufgaben, bei denen das Systemverhalten unbekannt ist, bietet die EWR eine potentielle Alternative. Dies trifft beispielsweise auf das Verbrennungsverfahren der in der Einführung beschriebenen SEC zu. Der Einfluss der Kraftstoffeinspritzung auf die Druckerhöhung in einem SEC-Rohr ist derart komplex, dass eine Beschreibung in einem Modell mit angemessener Komplexität nicht möglich ist.

Das Konzept der EWR kann für beliebige, dynamische Systeme verwendet werden. Eine allgemeine Einführung in die EWR ist in Ariyur und Krstic, 2003 zu finden. Da der regelungstechnische Fokus dieser Arbeit aber vor allem auf zyklischen Systemen liegt, wird in diesem Kapitel ausschließlich auf einen EWR-Ansatz eingegangen, der starke Ähnlichkeit zur ILR aufweist. Ausgehend von der in Khong, Nešić und Krstić, 2016 vorgestellten iterativen EWR wird in den folgenden Abschnitten eine iterative EWR präsentiert, die sowohl für Systeme mit allgemeinen, linearen Eingangsbeschränkungen als auch für ganzzahlige Beschränkungen in der Aktuation anwendbar ist. Bevor abschließend experimentelle Ergebnisse mit dieser EWR-Methode an einem SEC-Experiment gezeigt werden, wird zunächst in einer Simulationsstudie ein Vergleich zu einer modellgestützten ILR, wie sie in Kapitel 6 beschrieben ist, gezogen.

7.1 Extremwertregelung für zyklische Prozesse

Für die Betrachtung von zyklischen Prozessen wird im Folgenden ein unbekannter Zusammenhang

$$\mathbf{y}_k = \mathbf{f}^*(\boldsymbol{\mu}_k) \quad (7.1)$$

zwischen dem Systemeingang $\boldsymbol{\mu}_k$ und dem gemessenen Systemausgang \mathbf{y}_k im Zyklus k angenommen. Wie bei der normoptimalen ILR wird die Regelgüte in einem Zyklus k mit einer Kostenfunktion

$$J_k = J(\mathbf{y}_k) \quad (7.2)$$

in Abhängigkeit der Messung des Systemausgangs \mathbf{y}_k bewertet. Da der Systemausgang von der entsprechenden Stellgröße $\boldsymbol{\mu}_k$ abhängt, sind auch die Kosten eine Funktion der Stellgröße:

$$J_k = J(\mathbf{y}_k) = J(\mathbf{f}^*(\boldsymbol{\mu}_k)) = J^*(\boldsymbol{\mu}_k) \quad . \quad (7.3)$$

Da der Zusammenhang in Gl. (7.1) unbekannt ist, ist folglich auch die Funktion $J^*(\boldsymbol{\mu}_k)$ unbekannt. Hier zeigt sich der Unterschied zu einer ILR-Anwendung, wo ein Eingangs-/Ausgangsmodell

$$\mathbf{y}_k = \mathbf{f}^*(\boldsymbol{\mu}_k) \approx \mathbf{f}(\boldsymbol{\mu}_k) \quad (7.4)$$

7 Extremwertregelung

bekannt ist, um das reale Systemverhalten zu approximieren. Dadurch ist auch die Kostenfunktion

$$J_k = J(\mathbf{y}_k) = J(\mathbf{f}(\boldsymbol{\mu}_k)) = J'(\boldsymbol{\mu}_k) \quad (7.5)$$

ein bekannter algebraischer Ausdruck.

7.1.1 Extremwertsuche

Ziel der EWR ist wie bei der ILR die Stellgröße über die Zyklen anzupassen, um die Kostenfunktion kontinuierlich zu minimieren. Für die normoptimale ILR kann das Minimum der Kostenfunktion aus Gl. (7.5) für den nächsten Zyklus $k+1$ unter der Annahme zyklusinvarianter Störungen und Referenz analytisch berechnet werden. Es ergibt sich das Stellgesetz in Gl. (6.26) aus Abschnitt 6.2 in Abhängigkeit der Stellgröße des aktuellen Zyklus $\boldsymbol{\mu}_k$ und des entsprechenden Regelfehlers \mathbf{e}_k .

Wenn nun kein Modell zur Verfügung steht, muss die Systeminformation für die Optimierung aus vorhandenen Messdaten bestimmt werden. Dafür wird in der zyklischen EWR, wie in der konventionellen EWR, aus der Messung eine Gradienteninformation gewonnen. Abweichend von der konventionellen EWR muss im zyklischen Fall das System nicht als annähernd statisch angenommen werden, da die Abbildung für einen Zyklus tatsächlich statisch ist. Die unbekannte Kostenfunktion aus Gl. (7.3) kann im aktuellen Zyklus k durch ein Taylor-Reihe zweiter Ordnung

$$J^*(\boldsymbol{\mu}_k + \Delta\boldsymbol{\mu}) \approx J^*(\boldsymbol{\mu}_k) + \nabla J^*(\boldsymbol{\mu}_k)\Delta\boldsymbol{\mu} + \frac{1}{2}\Delta\boldsymbol{\mu}^T \mathbf{H}_{J^*}(\boldsymbol{\mu}_k)\Delta\boldsymbol{\mu} \quad (7.6)$$

angenähert werden. Dabei beschreibt $\boldsymbol{\mu}_k + \Delta\boldsymbol{\mu}$ eine Stellgröße mit der Abweichung $\Delta\boldsymbol{\mu}$ zur aktuellen Stellgröße $\boldsymbol{\mu}_k$.

Um in einem ersten Schritt den Gradienten $\nabla J^*(\boldsymbol{\mu}_k)$ mit Hilfe eines zentralen Differenzenquotienten zu approximieren, muss die benötigte Messinformation durch eine Perturbation der aktuellen Stellgröße $\boldsymbol{\mu}_k = [\mu_{k,1}, \dots, \mu_{k,i}, \dots, \mu_{k,n_u}]^T$ generiert werden. Dafür werden in den $2n_u$ Zyklen $\{k+1, \dots, k+2n_u\}$ nacheinander die perturbierten Stellgrößen

$$\boldsymbol{\mu}_{k+(i-1) \cdot 2+1} = \boldsymbol{\mu}_{k,i}^+ = [\mu_{k,1}, \dots, \mu_{k,i} + \delta_{k,i}^+, \dots, \mu_{k,n_u}]^T \quad (7.7)$$

$$\boldsymbol{\mu}_{k+(i-1) \cdot 2+2} = \boldsymbol{\mu}_{k,i}^- = [\mu_{k,1}, \dots, \mu_{k,i} - \delta_{k,i}^-, \dots, \mu_{k,n_u}]^T \quad (7.8)$$

mit

$$i \in \{1, \dots, n_u\}, \quad (7.9)$$

$$\delta_i^+, \delta_i^- \geq 0 \quad \forall i \quad (7.10)$$

als Ansteuerung auf das System angewendet. Damit ergeben sich in den entsprechenden Zyklen die Messungen des Systemausgangs

$$\mathbf{y}_{k+(i-1) \cdot 2+1} = \mathbf{y}_{k,i}^+ \quad (7.11)$$

$$\mathbf{y}_{k+(i-1) \cdot 2+2} = \mathbf{y}_{k,i}^- \quad , \quad (7.12)$$

aus denen wiederum mit dem bekannten Zusammenhang aus Gl. (7.2) die Kosten

$$J_{k,i}^+ = J(\mathbf{y}_{k,i}^+) \quad (7.13)$$

$$J_{k,i}^- = J(\mathbf{y}_{k,i}^-) \quad (7.14)$$

bestimmt werden können. Am Ende des Zyklus $k + 2n_u$ ist es dann möglich, eine Näherung des Gradienten

$$\nabla J^*(\boldsymbol{\mu}_k) \approx \begin{bmatrix} \frac{J_{k,1}^+ - J_{k,1}^-}{\delta_{k,1}^+ + \delta_{k,1}^-} \\ \vdots \\ \frac{J_{k,i}^+ - J_{k,i}^-}{\delta_{k,i}^+ + \delta_{k,i}^-} \\ \vdots \\ \frac{J_{k,n_u}^+ - J_{k,n_u}^-}{\delta_{k,n_u}^+ + \delta_{k,n_u}^-} \end{bmatrix} \quad (7.15)$$

für Gl. (7.6) zu bestimmen. Für den darauf folgenden Zyklus $k + 2n_u + 1$ kann nun die gewonnene Messinformation für die Optimierung der Stellfolge $\boldsymbol{\mu}_{k+2n_u+1}$ genutzt werden. Ein Optimierungsvorgang besteht folglich aus insgesamt $2n_u + 1$ Zyklen. Um die Notation im Folgenden zu vereinfachen, wird eine neue Zählvariable \bar{p} eingeführt, die den eigentlichen Optimierungsschritt von Zyklus k zu $k + 2n_u + 1$ markiert:

$$\boldsymbol{\mu}_{k=0} = \boldsymbol{\mu}_{\bar{p}=0} \quad (7.16)$$

$$\boldsymbol{\mu}_{k=2n_u+1} = \boldsymbol{\mu}_{\bar{p}=1} \quad (7.17)$$

$$\vdots$$

$$\boldsymbol{\mu}_{k=\bar{p} \cdot (2n_u+1)} = \boldsymbol{\mu}_{\bar{p}=\bar{p}^*} \quad (7.18)$$

Für die Approximation der zweiten Ableitung in Form der Hesse-Matrix besteht generell ebenfalls die Möglichkeit, die Einträge durch die Anwendung eines Differenzenquotienten numerisch zu bestimmen. Dazu wären allerdings zusätzliche Zyklen mit weiteren Perturbationen der Stellfolge notwendig, um die benötigte Messinformation zu generieren. Alternativ wird hier ein Ansatz gewählt, der sich an der Idee des bekannten BFGS-Algorithmus orientiert. Die Verwendung des BFGS-Algorithmus im Zusammenhang einer EWR wurde in Lange und King, 2020 vorgestellt. An dieser Stelle wird im speziellen die DFP-Methode als Teilschritt des BFGS-Verfahrens verwendet. Die Herleitung ist im Anhang B zu finden. Die Hesse-Matrix wird dabei iterativ über die Optimierungsschritte angepasst, um die Schätzung zu verbessern:

$$\hat{\mathbf{H}}_{J^*}(\boldsymbol{\mu}_{\bar{p}}) \approx (\mathbf{I} - \rho \mathbf{g}(\Delta \boldsymbol{\mu}_{\bar{p}-1})^T) \hat{\mathbf{H}}_{J^*}(\boldsymbol{\mu}_{\bar{p}-1}) (\mathbf{I} - \rho(\Delta \boldsymbol{\mu}_{\bar{p}-1}) \mathbf{g}^T) + \rho \mathbf{g} \mathbf{g}^T \quad (7.19)$$

Dabei gilt

$$\begin{aligned} \Delta \boldsymbol{\mu}_{\bar{p}-1} &= \boldsymbol{\mu}_{\bar{p}} - \boldsymbol{\mu}_{\bar{p}-1} \\ \mathbf{g} &= \nabla J^*(\boldsymbol{\mu}_{\bar{p}}) - \nabla J^*(\boldsymbol{\mu}_{\bar{p}-1}) \\ \rho &= \frac{1}{\mathbf{g}^T(\Delta \boldsymbol{\mu}_{\bar{p}-1})} \end{aligned} \quad .$$

Mit dem aus den Messdaten ermittelten Gradienten und der geschätzten Hesse-Matrix kann nun die Approximation aus Gl. (7.6) genutzt werden, um eine optimale Anpassungen der Stellfolge $\Delta \boldsymbol{\mu}_{\bar{p}} = \boldsymbol{\mu}_{\bar{p}+1} - \boldsymbol{\mu}_{\bar{p}}$ zwischen zwei Optimierungsschritten zu berechnen. Das Minimum der approximierten Kostenfunktion aus Gl. (7.6) wird im Zyklus $\boldsymbol{\mu}_{k=\bar{p} \cdot (2n_u+1)} = \boldsymbol{\mu}_{\bar{p}=\bar{p}^*}$ für

$$\Delta \boldsymbol{\mu}_{\bar{p}}^T = -\nabla J^*(\boldsymbol{\mu}_{\bar{p}})^T \hat{\mathbf{H}}_{J^*}(\boldsymbol{\mu}_{\bar{p}})^{-1} \quad (7.20)$$

erreicht. Damit ergibt sich für die Anpassung der Stellgröße von \bar{p} zu $\bar{p} + 1$ ein Stellgesetz

$$\boldsymbol{\mu}_{\bar{p}+1} = \boldsymbol{\mu}_{\bar{p}} + \Delta \boldsymbol{\mu}_{\bar{p}} = \boldsymbol{\mu}_{\bar{p}} - \left(\hat{\mathbf{H}}_{J^*}(\boldsymbol{\mu}_{\bar{p}})^{-1} \right)^T \nabla J^*(\boldsymbol{\mu}_{\bar{p}}) \quad , \quad (7.21)$$

das ähnlich zu dem Stellgesetz einer ILR aus Gl. (6.26) in Abschnitt 6.2 ist. Abweichend zur ILR führen die Zyklen, die für die Perturbation benötigt werden, jedoch zu einer deutlich langsameren Anpassung der Stellfolge.

7.1.2 Beschränkungen

Wie bereits mehrfach diskutiert, ist in einer realen Anwendung, im Speziellen im Bereich der aktiven Strömungsbeeinflussung, die Aktuation eines Systems oft beschränkt. In einem ersten Schritt werden Beschränkung betrachtet, die über lineare Ungleichungen ausgedrückt werden können. Bezogen auf die gesuchte Stellgröße $\mu_{\bar{p}+1}$ gilt

$$\Omega \mu_{\bar{p}+1} \leq c \quad . \quad (7.22)$$

Da sich die hier gewählte Formulierung der Kostenfunktion auf die Änderung der Stellgröße $\Delta \mu_{\bar{p}}$ bezieht, muss die Beschränkung aus Gl. (7.22) entsprechend transformiert werden, um die Beschränkung auf $\Delta \mu_{\bar{p}}$ zu beziehen:

$$\Omega (\mu_{\bar{p}} + \Delta \mu) \leq c \quad (7.23)$$

$$\Omega \Delta \mu \leq \underbrace{c - \Omega \mu_{\bar{p}}}_{c_T} \quad . \quad (7.24)$$

Dieser Schritt kann analog auch für Gleichungsbeschränkungen durchgeführt werden.

Zusätzlich zu einer linearen Beschränkung kann auch hier eine Aktuation in einem ganzzahligen Wertebereich berücksichtigt werden. Wie bei der ILR, muss dann ein ganzzahliges, quadratisches Optimierungsproblem, beispielsweise über eine *Branch-and-Bound*-Suche, gelöst werden. Um auch hier zwischen dem ganzzahligen und reellwertigen Fall zu unterscheiden, wird wieder die Notation \mathbf{u} für ganzzahlige Stellfolgen verwendet.

Quadratisches Programm (QP) 10: *Die optimale, ganzzahlige Anpassung der Stellfolge für den Optimierungsschritt $\bar{p} + 1$ unter Berücksichtigung der durch $\mathbf{H}_{J^*}(\mathbf{u}_{\bar{p}})$ und $\nabla J^*(\mathbf{u}_{\bar{p}})$ approximierten Kostenfunktion ergibt sich aus*

$$\Delta \mathbf{u}_{\bar{p}} = \arg \min_{\Delta \mathbf{u}} \quad \Delta \mathbf{u}^T \hat{\mathbf{H}}_{J^*}(\mathbf{u}_{\bar{p}}) \Delta \mathbf{u} + \nabla J^*(\mathbf{u}_{\bar{p}})^T \Delta \mathbf{u} \quad (7.25)$$

unter der Bedingung

$$\Delta \mathbf{u} \in \mathbb{N}^{n_u} \quad (7.26)$$

$$\Omega \Delta \mathbf{u} \leq c_T \quad . \quad (7.27)$$

Im Unterschied zur ganzzahligen ILR muss das QP 10 nur alle $2n_u + 1$ Zyklen gelöst werden. Steht nicht ausreichend Rechenzeit zur Verfügung, kann z.B. eine Verzögerung von einem Optimierungsschritt eingefügt werden, um die Zeit für die Lösung zu vergrößern.

7.2 Ergebnisse für 1D-Euler-Simulation

Bevor am Ende dieses Kapitels der beschriebene EWR-Ansatz auf einen experimentellen SEC-Aufbau angewendet wird, soll an dieser Stelle zunächst der Vergleich zu dem modellbasierten Ansatz der ILR gezogen werden. Um möglichst repräsentative Ergebnisse in Bezug auf die

Anwendung für die in der Einleitung beschriebene Problematik der Kraftstoffeinspritzung an einer SEC zu erhalten, wird hier ein Simulationsbeispiel betrachtet, welches die Einspritzung in ein entsprechendes Verbrennungsrohr vereinfacht nachbildet.

Das betrachtete System basiert wieder auf der in Abschnitt 3.1.4 beschriebenen 1-D Strömungssimulation auf Basis der Euler-Gleichungen. Es wird von einem 1 m langen Rohr mit 0.1 m Durchmesser ausgegangen, in welches vor dem Einlass Kraftstoff eingespritzt werden kann. Vereinfacht wird in diesem Beispiel von Luft mit ausschließlich den Bestandteilen Sauerstoff (21 %) und Stickstoff (79 %) ausgegangen. Als Kraftstoff wird Wasserstoff verwendet. Die Zusammensetzung des Gemischs an der Stelle x im Rohr wird durch das Äquivalenzverhältnis

$$\Phi(x) = \frac{n_{H_2}(x)}{(1 - n_{H_2}(x)) \cdot 0.21 \cdot 2} \quad (7.28)$$

definiert. Hier ist n_{H_2} der Stoffmengenanteil von Wasserstoff. Für die Stoffmengenanteile von Sauerstoff und Stickstoff gilt entsprechend

$$n_{O_2}(x) = 0.21 \cdot (1 - n_{H_2}(x)) \quad (7.29)$$

$$n_{N_2}(x) = 0.79 \cdot (1 - n_{H_2}(x)) \quad (7.30)$$

Die Einspritzung wird über eine veränderliche Randbedingung am Einlass des Rohres realisiert. Fällt der Druck unter die Umgebungsbedingungen, strömt Gas in das Rohr. Liegt der Druck am Rand des Rohres über dem Umgebungsdruck wird ein reflektierender Rand angenommen. Am Auslass wird eine Expansion auf Umgebungsbedingungen simuliert. Für die Umgebung gelten atmosphärische Bedingungen. In Anlehnung an den weiter unten diskutierten, experimentellen SEC-Versuchsaufbau, wird von einer Einspritzvorrichtung mit diskretem Charakter ausgegangen. Für die Aktuierung, die im diskreten Zeitschritt l durch das entsprechende Äquivalenzverhältnis am Einlass repräsentiert wird, gilt die Beschränkung

$$\mu_l \in [0, 2.0] \quad (7.31)$$

Ähnlich zu der technischen Umsetzung der Einspritzung mit einer festen Anzahl an schaltenden Ventilen wird eine ganzzahlige Stellgröße

$$u_l \in \{0, 1, \dots, 20\} \quad (7.32)$$

eingeführt. Das tatsächliche Äquivalenzverhältnis aus Gl. (7.31) kann dann über die Skalierung

$$\mu_l = \frac{1}{10} u_l \quad (7.33)$$

bestimmt werden. Es wird eine Feuerfrequenz von $f_f = 100$ Hz betrachtet. Ein Zyklus beginnt mit der Verbrennung und endet mit der Einspritzung. Die Einspritzphase beschränkt sich auf den Zeitbereich $t_I \in [4.5, 9.7]$ ms. In diesem Zeitbereich fällt in der betrachteten Simulation der Druck am Einlass des Rohres unter den Umgebungsdruck und eine Befüllung ist möglich. Die Einspritzung ist in 20 zeitlich äquidistante Schritte unterteilt.

Wie in der Einleitung beschrieben, ist der strömungsmechanische Vorgang der SEC durch die Kopplung der akustischen und reaktiven Zeitkonstanten sehr komplex und wird von verschiedenen Faktoren beeinflusst. Welche Form der Einspritzkurve zu einem stabilen SEC-Betrieb führt, ist aktuell noch Gegenstand der Forschung (Yücel, Habicht, Arnold et al., 2021). Dies ist vor allem auf den äußerst schwierig zu modellierenden Zusammenhang zwischen der

Kraftstoffkonzentration am Einlass und der tatsächlichen Druckerhöhung zurückzuführen. Die Druckerhöhung eines Zyklus kann beispielsweise über das zeitliche Integral des Drucks am Auslass bewertet werden. Da die Definition eines passenden Einspritzprofils nicht Teil dieser Arbeit ist, wird im Folgenden davon ausgegangen, dass auf Basis von umfangreichen Untersuchungen zu verschiedenen Kraftstoffen und den Strömungsbedingungen im Rohr eine optimale räumliche Kraftstoffverteilung festgelegt werden kann. Im späteren Verlauf dieses Kapitels wird abweichend davon auch eine Optimierung der Drucksignale betrachtet. Die Verbrennung wird in jedem Zyklus idealisiert angenommen, mit einem gleichmäßigen Druckanstieg auf 6 bar in den ersten 40 % des Verbrennungsrohres. Für das Simulationsbeispiel wird als Systemausgang \mathbf{y}_k die Kraftstoffkonzentration am Ende der Einspritzung an 20 räumlich äquidistanten Stellen in einem Bereich $x_I \in [0.04, 0.23]$ gewählt.¹

Als Referenz für das Verhalten der ganzzahligen EWR wird eine ganzzahlige ILR herangezogen, die mit der in Abschnitt 6.3.2 eingeführten OMS ausgelegt wurde. Dafür wird zunächst ein statisches Modell

$$\hat{\mathbf{y}}_k = \mathbf{G}\mathbf{u}_k \quad (7.34)$$

identifiziert, welches die ganzzahlige Repräsentation der Füllkurve $\mathbf{u}_k \in \mathbb{N}^{20}$ auf die Kraftstoffkonzentration $\hat{\mathbf{y}}_k \in \mathbb{R}^{20}$ für einen Einspritzzyklus k abbildet. Zur Identifikation des Modells wird das System mit 40 zufälligen Stellfolgen angeregt und die Systemmatrix \mathbf{G} über eine Optimierung bestimmt, welche die Fehlerquadrate minimiert:

$$\mathbf{G} = \arg \min_{\mathbf{G}} \sum_{i=1}^{40} (\mathbf{y}_i - \hat{\mathbf{y}}_i)^T (\mathbf{y}_i - \hat{\mathbf{y}}_i) = \sum_{i=1}^{40} (\mathbf{y}_i - \bar{\mathbf{G}}\mathbf{u}_i)^T (\mathbf{y}_i - \bar{\mathbf{G}}\mathbf{u}_i) \quad . \quad (7.35)$$

Die Kardinalität der Menge der zulässigen Schritte \mathcal{V} für die OMS wird auf $|\mathcal{V}| = n_S = 100$ festgelegt, siehe NLP 1.

Die ganzzahlige EWR wird für dieses Beispiel mit einer Hesse-Matrix $\mathbf{H} = 0.1 \cdot \mathbf{I}$ initialisiert. Die Perturbation in jedem Schritt wird mit der Schrittweite $\delta_+ = \delta_- = 1$ durchgeführt. Ziel der Regelung ist es in diesem einfachen Beispiel, ein linear über die Länge des Rohres abfallendes Kraftstoffprofil einzustellen. Entsprechend wird die Referenz zu

$$r(x) = -5.21 \cdot x + 1.21 \quad x \in [0.04, 0.23] \quad (7.36)$$

gewählt. Damit gilt $r(0.04) \approx 1$ und $r(0.23) \approx 0$. Die initiale Stellfolge ist in beiden Ansätzen eine konstante Stellfolge von

$$\mathbf{u}_0 = 10 \cdot \mathbf{1}^{20 \times 1} \quad , \quad (7.37)$$

die einem konstanten Äquivalenzverhältnis von $\Phi = 1$ für den gesamten Zeitraum der Einspritzung entspricht. Hier ist $\mathbf{1}^{20 \times 1}$ ein stehender Vektor aus 20 Einsen.

Abbildung 7.1 zeigt den Verlauf der Fehlernormen über die Iterationen. Hier ist deutlich zu erkennen, dass sich der Fehler im modellbasierten Ansatz der ILR deutlich schneller abbaut als bei der Verwendung der EWR. Während der EWR-Ansatz erst nach 40 Zyklen, in

¹In einer realen Anwendung müsste für eine Verbrennung in den ersten 40 % des Rohres auch die ersten 40 % des Rohres mit Kraftstoff befüllt werden. Dies ist allerdings nur für bestimmte Rohrlängen und Einströmbedingungen möglich, die stark vom letztendlichen Verbrennungsprozess abhängen. Für eine einfache Realisierung und eine Fokussierung auf den Regler wird hier abweichend von den genannten Daten ausgegangen.

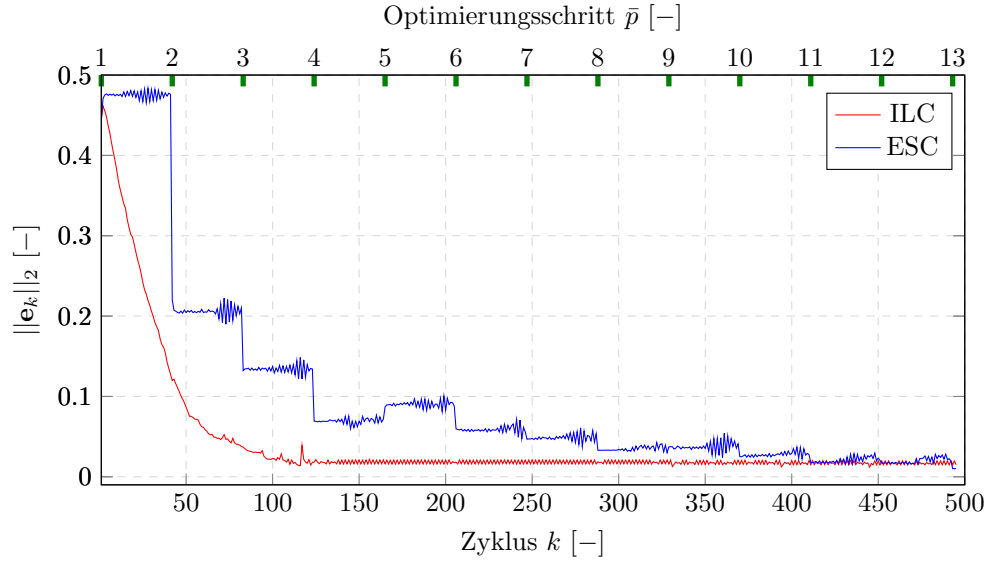


Abbildung 7.1: Verlauf der 2-Norm des Fehlers mit einer ganzzahligen ILR auf Basis der OMS und einer ganzzahligen EWR. Die Optimierungsschritte der zyklischen EWR sind in grün gekennzeichnet.

denen die Stellfolge perturbiert wird, Informationen gesammelt hat, die einen ersten Optimierungsschritt zulassen, kann die ILR die Fehlernorm im gleichen Zeitraum bereits um den Faktor vier reduzieren. Der ILR-Ansatz konvergiert bereits nach etwas über 100 Zyklen während der EWR-Ansatz erst nach über 400 Iteration einen annähernd konvergierten Zustand erreicht. Es ist jedoch festzuhalten, dass auch der modellfreie Ansatz ein ähnliches Fehlerniveau erreicht wie der modellbasierte Ansatz. Zudem ist für beide Ansätze das nichtmonotone Konvergenzverhalten anzumerken, wobei der ILR-Ansatz auch hier leichte Vorteile zeigt.²

Der Fehlerverlauf des EWR-Ansatzes deutet daraufhin, dass die zweite Hälfte der Einspritzphase deutlich stärkeren Einfluss hat als die erste Hälfte. Dies ist deutlich in den stärkeren Ausschlägen in Folge der Perturbationen der einzelnen Einträge der Stellfolge zu erkennen. Mit dem EWR-Ansatz muss ein solcher Zusammenhang erst aus den Messdaten erlernt werden. Im Gegensatz dazu ist dieser Aspekt in dem Modell bereits enthalten und führt zu Beginn zu zielgenaueren Einstellung des Einspritzprofils mit dem ILR-Ansatz.

Obwohl die beiden Ansätze wie beschrieben auf ein ähnliches Fehlerniveau führen, ergeben sich doch sehr unterschiedliche Stellfolgen im konvergierten Zustand. Abbildung 7.2 zeigt die konvergierten Stellfolgen der Regelungsansätze sowie die daraus resultierende Kraftstoffkonzentration am Rohreinlass über die Zeit der Einspritzung. Die Strömungsbedingungen am Einlass des Rohres sind durch die Strömungsgeschwindigkeit illustriert. Es ist zu erkennen, dass der EWR-Ansatz einen etwas gleichmäßigeren Verlauf über die Zeit einstellt als der ILR-Ansatz. Trotz der stark abweichenden Einspritzprofile ergeben sich sehr ähnliche räumliche Verteilungen des Kraftstoffs wie in Abb. 7.3 zu sehen ist. Mit beiden Ansätzen kann die Referenz annähernd genau getroffen werden. Ein Unterschied zwischen den beiden erzielten Verläufen ist kaum zu erkennen.

Dieses einfache Simulationsbeispiel zeigt bereits die deutlichen Unterschiede zwischen den beiden Ansätzen und verdeutlicht Vor- und Nachteile des vorgestellten EWR-Ansatzes. Lie-

²Die OMS für die ILR führt hier nicht zu einem monotonen Konvergenzverhalten, da die Nichtlinearität des Systems stark ausgeprägt ist. Eine zyklusinvariante Approximation durch ein Modell der Form $y_k = \mathbf{G}u_k + \mathbf{b}$ ist nicht exakt möglich.

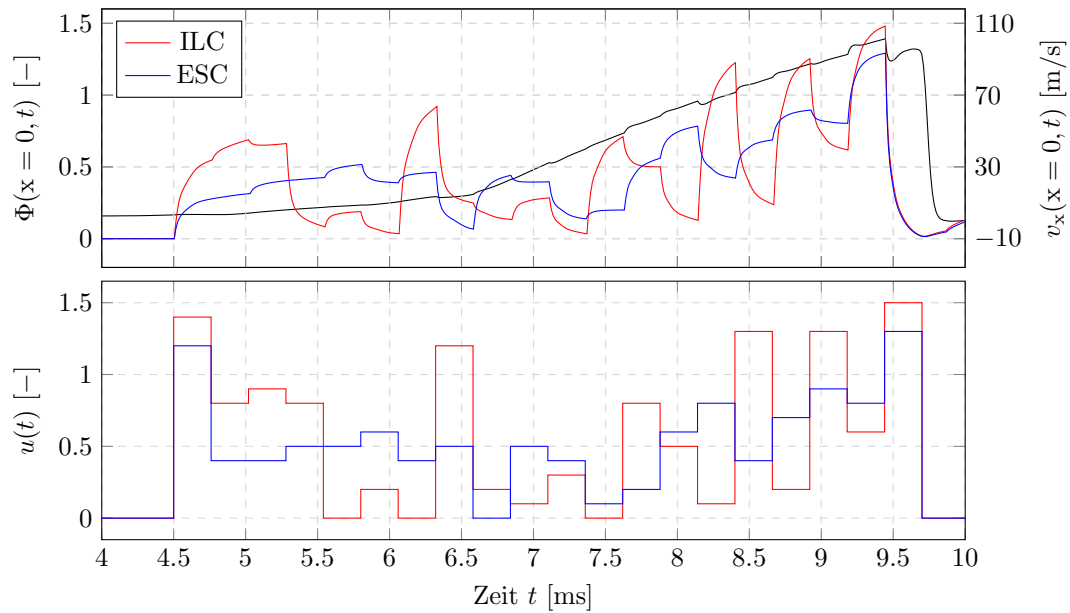


Abbildung 7.2: Oben: Kraftstoffkonzentration an der Stelle $x = 0$ über die Zeit der Einspritzung. Zusätzlich ist in schwarz die Geschwindigkeit ebenfalls am Einlass dargestellt. Unten: Eingestellte Stellfolge über die Zeit der Einspritzung. Die Stellgröße entspricht der Kraftstoffkonzentration vor dem Rohr.

gen ausreichend gute Modellinformationen vor, ist der modellbasierte Ansatz der ILR der EWR klar überlegen. Ein Vorteil auf den an dieser Stelle nochmal hingewiesen sei, ist das deterministische Rechenverhalten der ILR aus einer OMS. Der Rechenaufwand des Optimierungsschritts in der EWR ist nicht exakt abzuschätzen. In diesem Beispiel ist dieser Faktor weniger relevant, da zwischen dem Ende der Einspritzphase und dem Beginn der nächsten Einspritzphase eine Pause ohne Aktuation existiert, die für die Berechnung genutzt werden kann. In allgemeinen Fällen kann der unbekannte Rechenaufwand jedoch zu einem Problem werden. Zudem kann das langsame Lernverhalten der EWR bei Prozessen mit schnellen Störungen problematisch sein. Im Fall einer plötzlich auftretenden Störung kann es mitunter mehrere Optimierungsschritte mit entsprechenden Perturbationszyklen dauern, bevor die Regelung das System wieder in einen zufriedenstellenden Fehlerbereich zurückführt. Abschließend ist jedoch festzuhalten, dass die ganzzahlige EWR eine sinnvolle Alternative für zyklische Prozesse darstellen kann, wenn eine mathematische Modellierung schwierig oder nicht möglich ist.

7.3 Extremwertregelung an einem SEC-Experiment

Die experimentelle Umsetzung einer SEC ist technisch äußerst komplex und aktuell noch Gegenstand der Forschung. Ein stabiler zyklischer Betrieb konnte bisher noch nicht erreicht werden. Der hier diskutierte Aufbau wurde vor allem zur Untersuchung grundsätzlicher Verbrennungseigenschaften der SEC entwickelt. Im Speziellen liegt der Fokus der Untersuchungen auf den Möglichkeiten, über die Kraftstoffsichtung die Verbrennung zuverlässig zu beeinflussen. In der experimentellen Umsetzung ist im Gegensatz zu dem gezeigten Simulationsbeispiel das Ziel, direkt die Drucksignale im Verbrennungsrohr zu optimieren.

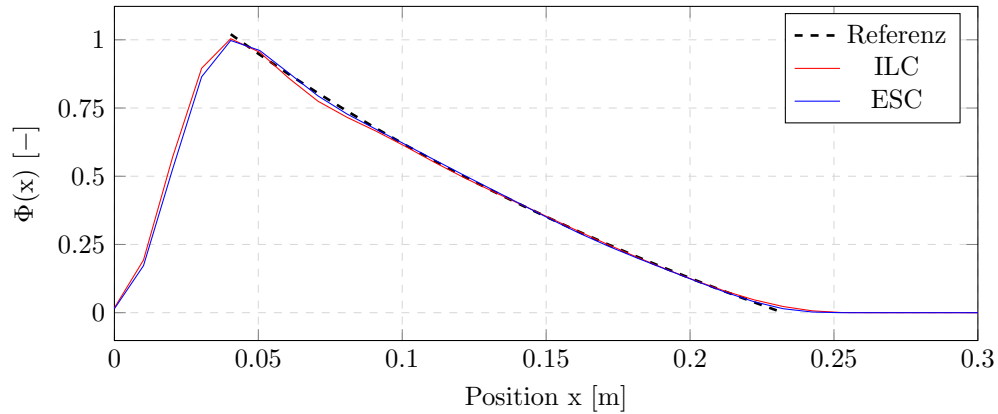


Abbildung 7.3: Räumliche Kraftstoffverteilung, die sich mit den konvergierten Stellfolgen aus Abbildung 7.2 ergeben.

7.3.1 Experimenteller Aufbau

In den letzten Jahren wurde die SEC in verschiedenen Konfigurationen experimentell untersucht. Dabei beziehen sich alle bisherigen Untersuchungen auf einen Aufbau bei atmosphärischen Druckbedingungen. Daraus folgt, dass die im Mittel realisierbaren Zündverzugszeiten im Vergleich zu den akustischen Zeitkonstanten deutlich zu groß sind, um einen zyklischen Betrieb, wie er in Abschnitt 1.1 erläutert wurde, bei diesen Bedingungen und der angenommenen Rohrlänge zu ermöglichen. Die experimentellen Arbeiten sind bisher vor allem auf die Beeinflussung des Zündverhaltens konzentriert.

In einem ersten Aufbau, wie er in Reichel et al., 2016 und Yücel, Völzke und Paschereit, 2019 beschrieben ist, wurden die langen Zündverzugszeiten durch einen Bypass kompensiert, in den die Strömung nach der Einspritzung umgeleitet wird. Dadurch kommt die Strömung im eigentlichen Verbrennungsrohr zum Stehen und das Kraftstoff-Luft-Gemisch kann bis zur Zündung im Rohr verweilen. Danach kann die Strömung wieder in das Verbrennungsrohr zurück geschaltet werden, um das Rohr zu spülen, bevor der nächste Zyklus beginnen kann. Die Problematik dieses Aufbaus liegt in dem Schaltvorgang zwischen Bypass und Verbrennungsrohr. Sowohl im Aufbau mit einem fluidischen Schalter, wie in Reichel et al., 2016, als auch im Aufbau mit an den Rohrenden montierten Wastegates in Yücel et al., 2019 beeinflusste der Schaltprozess die Strömungsbedingungen im Verbrennungsrohr stark und damit auch die Kraftstoffschichtung. Eine Modellierung ist für dieses System nur schwer möglich, wie die Arbeit in Arnold, King, Yücel, Völzke und Paschereit, 2019 zeigt. Zudem ist der Aufbau mit einem Bypass deutlich schwerer zu isolieren, was zu großen Wärmeverlusten führt und dadurch das Zündverhalten zusätzlich negativ beeinflusst.

Durch Untersuchungen zum Zündverhalten des Kraftstoffs und eine deutlich dynamischere Einspritzung ist es möglich, die mittlere Zündverzugszeit deutlich zu reduzieren. Damit kann auf einen Bypass verzichtet werden. Die Zündverzugszeit wird deshalb im aktuellen Aufbau alternativ mit einem Konvektionsrohr kompensiert. Ein Schema des in dieser Arbeit verwendeten experimentellen Aufbaus ist in Abb. 7.4 dargestellt. Der Versuch entspricht dem in Yücel, Habicht, Arnold et al., 2021 beschriebenen Aufbau. Die verbleibende Zündverzugszeit wird durch ein dem eigentlichen Verbrennungsrohr vorgeschaltetes Konvektionsrohr kompensiert. Davor befindet sich die Einspritzgeometrie, an die zehn Schaltventile mit einer Schaltfrequenz von maximal $f = 200$ Hz angeschlossen sind. Der Vordruck der Ventile ist über einen kommerziellen Domdruckregler einstellbar und beträgt im Betrieb 5 bar. Zur Abführung des

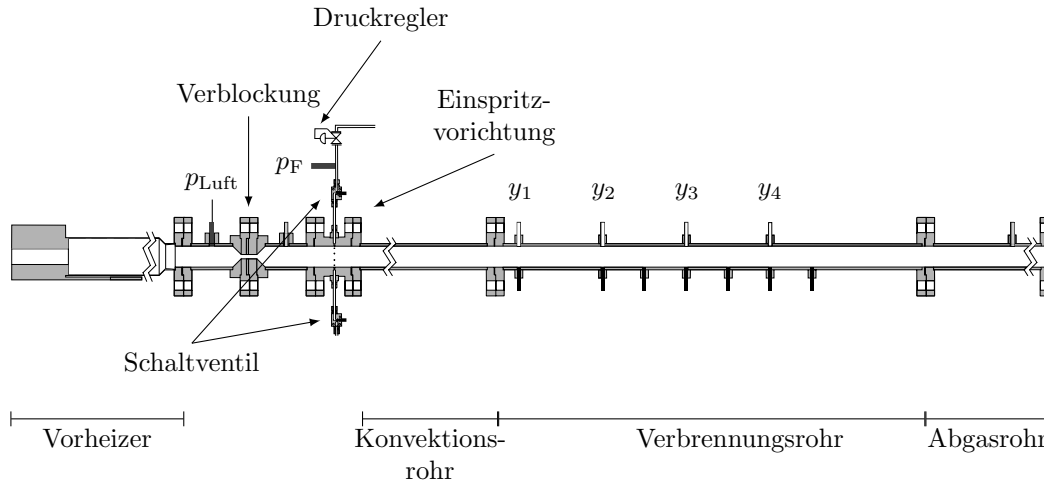


Abbildung 7.4: Experimenteller Aufbau eines SEC-Verbrennungsrohres (Yücel, Habicht, Arnold et al., 2021)

verbrannten Gases ist der eigentlichen Brennkammer ein Abgasrohr nachgeschaltet.

Der gesamte Aufbau wird mit einem konstanten Luftmassenstrom von $\dot{m} = 30 \text{ kg/h}$ gespeist, der in einem Vorheizer auf eine Temperatur von $\vartheta = 800 \text{ K}$ vorgeheizt wird. Um möglichst konstante Strömungsbedingungen im Rohr zu gewährleisten, ist der Querschnitt hinter dem Vorheizer verblockt. Neben den Druckmessungen vor der Verblockung und vor der Einspritzung sind hochdynamische Drucksensoren im Verbrennungsrohr verbaut, die Drucksignale bis zu 10 kHz zuverlässig aufnehmen können. Weitere technische Details zum Versuchsaufbau sind in Yücel, Habicht, Arnold et al., 2021 und Yücel, Habicht, Jaeschke et al., 2021 zu finden.

7.3.2 Ergebnisse

Für den experimentellen Aufbau der SEC ist an dieser Stelle das Ziel, die Einspritzung über die Zeit so einzustellen, dass der Druckanstieg und damit die nutzbare Energie aus der Verbrennung maximiert wird. Da die Aktuation wie beschrieben über 10 einzelne Schaltventile realisiert ist, muss die Stellgröße $u(t)$ im ganzzahligen Wertebereich definiert werden, um Einflüsse einer Rundung zu vermeiden. Die Feuerfrequenz wird mit Rücksicht auf die Beschränkungen durch die atmosphärischen Bedingungen auf $f_f = 5 \text{ Hz}$ festgelegt. Damit ergibt sich eine Periodendauer von $T_p = 200 \text{ ms}$. Da allerdings nur zu Beginn eines Zyklus Kraftstoff für eine feste Zeit von $T_{pI} = 30 \text{ ms}$ eingespritzt wird, wird auch nur dieser Zeitbereich für die Regelung betrachtet. Die Aktuationsfrequenz wird mit Rücksicht auf die limitierte Schaltbandbreite der Ventile auf $f_u = 1/0.006 \text{ Hz}$ festgelegt, sodass die Stellfolge in einem Zyklus k durch

$$\mathbf{u}_k = [u_1, u_2, u_3, u_4, u_5]^T \quad (7.38)$$

definiert ist.

Der Druck wird an vier äquidistanten Stellen $x_M \in \{1, 2, 3, 4\}$ in der Brennkammer mit der Abtastfrequenz $f_y = 10 \text{ kHz}$ gemessen. Aufgrund zufälliger Schwankungen in Luft- und Kraftstoffzufuhr ist die Schuss-zu-Schuss-Varianz relativ groß im Verhältnis zu der Sensitivität des Systems auf die Aktuation. Daher ist eine Optimierung im Hinblick auf einzelne Schüsse nicht möglich. Um die Schwankungen zu kompensieren, werden immer 30 aufeinanderfolgende

Schüsse mit der gleichen Stellfolge \mathbf{u}_k durchgeführt und die resultierenden 30 Messungen werden gemittelt, um den Systemausgang

$$\mathbf{y}_k^{\mathbf{x}_M} = \frac{1}{30} \sum_{i=1}^{30} \mathbf{y}_{k,i}^{\mathbf{x}_M} \quad (7.39)$$

für den Zyklus k an der Stelle \mathbf{x}_M zu erhalten.

Die physikalische Beschränkung der Stellgröße, die aus der festen Anzahl an Ventilen resultiert, kann über die lineare Ungleichung

$$\begin{bmatrix} \mathbf{I}^5 \\ -\mathbf{I}^5 \end{bmatrix} \mathbf{u}_k \leq \begin{bmatrix} \mathbf{1}^{5 \times 1} \cdot 10 \\ \mathbf{0}^{5 \times 1} \end{bmatrix} \quad (7.40)$$

beschrieben werden. Dabei ist \mathbf{I}^5 eine Einheitsmatrix der Dimension fünf, $\mathbf{1}^{5 \times 1}$ und $\mathbf{0}^{5 \times 1}$ sind Vektoren entsprechender Größe mit Einsen bzw. Nullen. Zusätzlich soll vermieden werden, dass der Druckanstieg im Verbrennungsrohr auf eine Variation der gesamten Kraftstoffmenge während eines Zykluses zurückzuführen ist. Nur so kann sichergestellt werden, dass der Anstieg des Drucks ausschließlich aus der Anpassung der Kraftstoffsichtung resultiert. Dafür wird die Summe der offenen Ventile ebenfalls über eine lineare Ungleichung

$$\begin{bmatrix} \mathbf{1}^{1 \times 5} \\ -\mathbf{1}^{1 \times 5} \end{bmatrix} \mathbf{u}_k \leq \begin{bmatrix} 40 \\ 30 \end{bmatrix} \quad (7.41)$$

beschränkt. Durch diese Beschränkung wird der Bereich, in dem die gesamte Kraftstoffmenge über einen Zyklus liegen kann, deutlich eingeschränkt. Eine feste Summe wird nicht gewählt, um einen gewissen Spielraum für die Anpassung durch die EWR zu behalten. Die Beschränkungen aus Gl. (7.40) und Gl. (7.41) können zusammengefasst werden in eine Ungleichung

$$\underbrace{\begin{bmatrix} \mathbf{I}^5 \\ -\mathbf{I}^5 \\ \mathbf{1}^{1 \times 5} \\ -\mathbf{1}^{1 \times 5} \end{bmatrix}}_{\Omega} \mathbf{u}_k \leq \underbrace{\begin{bmatrix} \mathbf{1}^{5 \times 1} \cdot 10 \\ \mathbf{0}^{5 \times 1} \\ 40 \\ 30 \end{bmatrix}}_c, \quad (7.42)$$

die dann entsprechend Gl. (7.24) für die einzelnen Optimierungsschritte transformiert werden kann. Der Druckanstieg wird über die mittleren Maxima an allen vier Messstellen über alle 30 Schüsse eines Zykluses quantifiziert. Dafür werden alle Druckspitzen aufsummiert. Es ergibt sich die Kostenfunktion

$$J^*(\mathbf{u}_k) = \sum_{\mathbf{x}_M=1}^4 \sum_{i=1}^{30} \left\| \mathbf{y}_{k,i}^{\mathbf{x}_M}(\mathbf{u}_k) \right\|_{\infty} \cdot \frac{1}{30} \quad (7.43)$$

in Abhängigkeit der gemessenen Drucksignale. Hier bezeichnet $\|\cdot\|_{\infty}$ die Unendlichnorm.

In Abb. 7.5 ist der Verlauf der über die Schüsse gemittelten Druckspitzen über die Zyklen gezeigt. Es ist zu erkennen, dass an allen Messpositionen die Druckspitzen über die Optimierungsschritte deutlich vergrößert werden. Ab etwa dem zehnten Optimierungsschritt wird das erreichte Niveau gehalten. In Abb. 7.6 sind die Störungen in der Luft und der Kraftstoffversorgung dargestellt. Trotz dieser Störungen gelingt es mit der zyklischen EWR, den Druckanstieg im Rohr deutlich zu verbessern und zu halten. Der zugehörige Verlauf der Stellgröße ist in Abb. 7.7 gezeigt. Neben der Perturbation ist hier zu erkennen, dass die Stellfolge

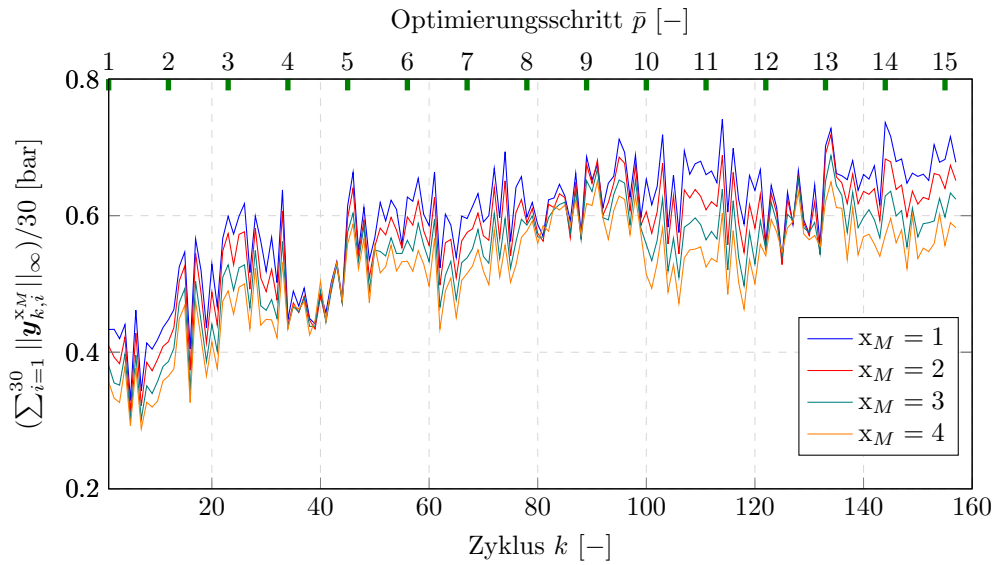


Abbildung 7.5: Mittlere Druckspitzen in einem Zyklus an allen vier Messpositionen $x_M \in \{1, 2, 3, 4\}$ in einem SEC-Verbrennungsrohr mit einer EWR. Die Optimierungsschritte der EWR sind in grün markiert.

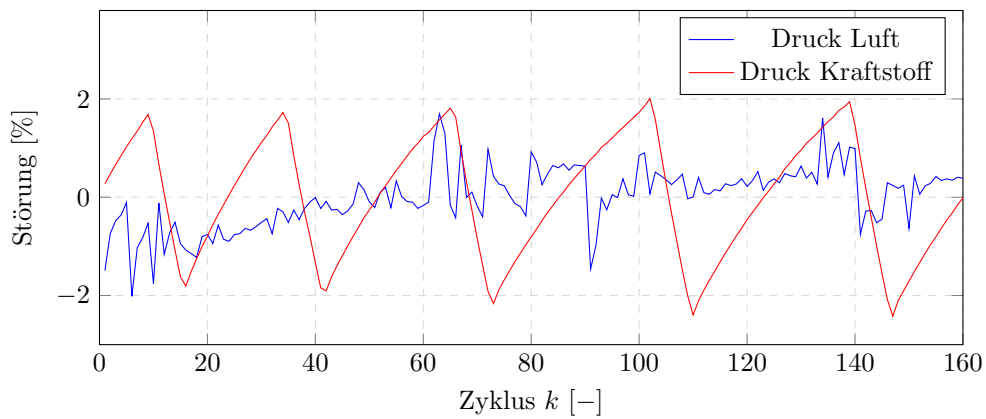


Abbildung 7.6: Verlauf der mittleren, prozentualen Störung des Drucks in der Luft- bzw. Kraftstoffversorgung über die Zyklen. Die Messpositionen sind in Abb. 7.4 dargestellt.

ab dem zehnten Optimierungsschritt sich im Wesentlichen nur noch für $i = 5$ ändert. Dieses Ergebnis deutet darauf hin, dass im auskonvergierten Zustand die angesprochenen Störungen in den Zuleitungen ausschließlich über den Kraftstoffanteil kompensiert werden können, der am Ende des Einspritzintervalls in das Verbrennungsrohr eingebracht wird.

Um einen noch besseren Eindruck zu gewinnen, wie der Regler auf den Druckanstieg wirkt, sind in Abb. 7.8 die Stellfolgen, die sich in Optimierungsschritt $\bar{p} = 1$, $\bar{p} = 6$ und $\bar{p} = 15$ ergeben, zusammen mit den über die Messstellen gemittelten Druckverläufen für einen einzelnen Schuss gezeigt. Es ist auch hier zu erkennen, dass mit der EWR die Druckspitzen über die Optimierungsschritte kontinuierlich vergrößert werden. Für eine größere Druckspitze ist eine Anpassung der Zündzeit notwendig, was sich in der Verschiebung der Druckverläufe zeigt. Dies ist ein komplexer, systemspezifischer Zusammenhang, der durch die EWR erlernt werden kann.

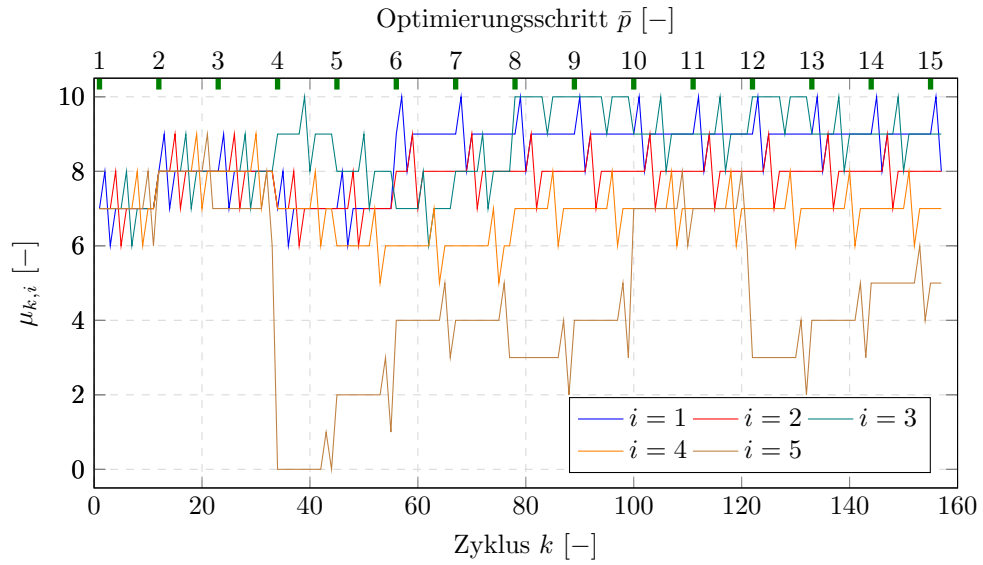


Abbildung 7.7: Aktuation der EWR über die Zyklen. Die Optimierungsschritte sind in grün gekennzeichnet.

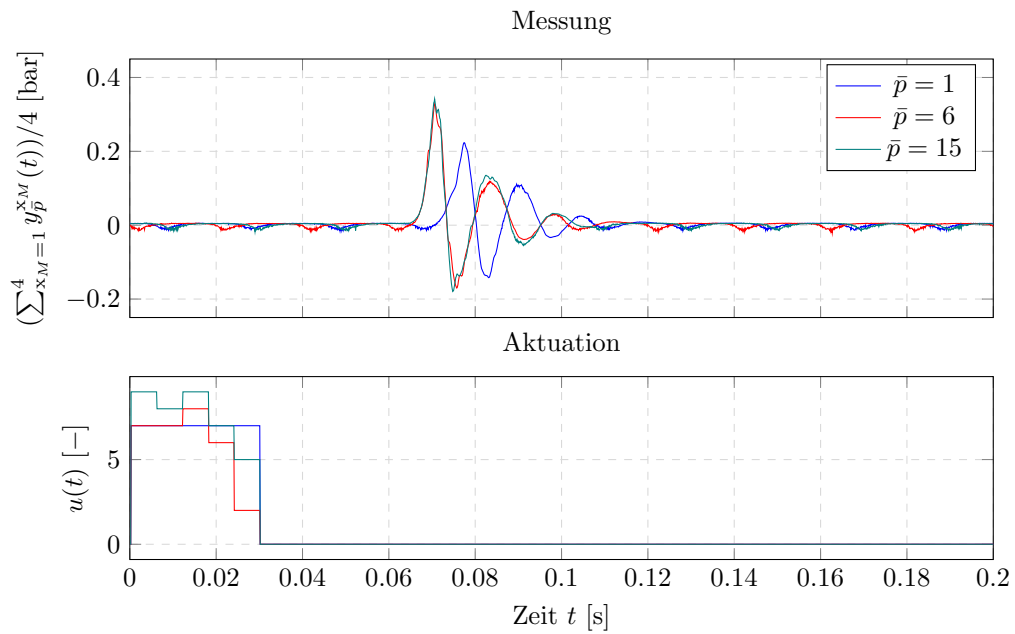


Abbildung 7.8: Oben: Zeitlicher Verlauf der über alle vier Messpositionen gemittelten relativen Druckverläufe für die Optimierungsschritte $\bar{p} \in \{1, 6, 15\}$. Unten: Zeitlicher Verlauf der Aktuation für die entsprechenden Optimierungsschritte.

8 Fazit

Die in dieser Arbeit vorgestellten Methoden liefern einen Beitrag zu zwei technisch relevanten Themen der aktiven Strömungsbeeinflussung. Zum einen wurde für die Modellierung von komplexen Strömungssystemen mit Echtzeitanforderungen eine Methode vorgestellt und für zwei Anwendungsbeispiele sowohl im Kontext einer Regelung als auch der Zustandsschätzung untersucht. Zum anderen wurden verschiedene Ansätze für den Umgang mit Eingangsbeschränkungen für zyklische Systeme diskutiert und simulativ bzw. experimentell getestet. An dieser Stelle sollen die gewonnenen Erkenntnisse kritisch bewertet und Kernaussagen hervorgehoben werden, bevor abschließend ein Ausblick für die einzelnen Methoden gegeben wird.

Modellierung

Der vorgestellte Ansatz einer Zustandsraummodellierung über den PINN-Ansatz hat in dieser Arbeit vor allem für Burgers-Gleichung zu guten Ergebnissen geführt. Dabei war eine zufriedenstellende Modellanpassung bereits mit einer relativ kleinen Anzahl an Trainingsdaten möglich. Das Training mit den Informationen aus der partiellen Differentialgleichung führt zudem zu besonders zuverlässigen Ergebnissen, auch für Eingangsdaten, die in der Anpassung des Modells nicht verwendet wurden. Zudem hat sich gezeigt, dass der Rechenaufwand der Modellauswertung so gering ist, dass eine Berechnung in Echtzeit problemlos möglich ist. Sowohl das Beispiel mit einem erweiterten Kalman-Filter als auch die Betrachtungen in einem MPC-Kontext zeigen, wie gut das aufgebaute Modell in praktischen Anwendungen funktioniert.

Für die Untersuchungen mit den Euler-Gleichungen sind die Ergebnisse deutlich kritischer zu bewerten. Es konnte auch für die Euler-Gleichungen ein Zustandsraummodell aufgebaut werden, mit dem in der Zustandsschätzung und der Regelung zufriedenstellende Ergebnisse erzielt werden konnten. Dafür waren allerdings diverse problemspezifische Anpassungen, wie der in Abschnitt 3.4.2 beschriebene zweistufige Trainingsprozess, notwendig. Zudem erforderte das Training eine sehr große Menge an Trainingsdaten, was zu einem enormen Zeitaufwand für die Optimierung des Netzes führt. Trotz diesen Bemühungen waren viele Versuche notwendig, um passende Parameter für den Trainingsprozess zu finden und am Ende ein gut angepasstes Netz zu erhalten. Ein allgemeines Vorgehen konnte daraus allerdings nicht abgeleitet werden. Zudem muss festgehalten werden, dass das Zustandsraummodell für die Euler-Gleichungen nicht in gleichem Maße robust ist wie das Modell für die Burgers-Gleichung. Zuverlässige Ergebnisse können nur in einem gewissen Bereich der Zustände erzielt werden, der in den Trainingsdaten gut repräsentiert ist. Trotzdem kann festgehalten werden, dass es gelungen ist, ein Zustandsraummodell aufzubauen, das bei relativ kleinem Rechenaufwand für die Auswertung gute Ergebnisse für ein Zustandsschätz- und Regelungsbeispiel ermöglicht hat. Generell scheint der Ansatz also das Potential zu haben, derart komplexe Systeme abzubilden.

Beschränkte Regelung

Die gezeigten Ergebnisse zur ganzzahlig beschränkten Regelung zeigen den enormen Vorteil, der sich ergeben kann, wenn die entsprechenden Beschränkungen direkt in der Optimierung in der Regelung berücksichtigt werden. Dies gilt insbesondere, wenn nur eine geringe Aktualisationsfrequenz im Verhältnis zu Systemdynamik technisch umsetzbar ist oder die Auflösung der Eingangsdiskretisierung sehr klein ist. Ein Extremfall für Letzteres stellt die in Abschnitt 6.4 ausführlich diskutierte, binäre Eingangsbeschränkung dar.

Das vorgestellte Syntheseverfahren der optimalen Menge für eine ganzzahlige ILR bietet eine Möglichkeit, für zyklische Prozesse eine besonders recheneffiziente Regelung aufzubauen, welche ganzzahlige und lineare Eingangsbeschränkungen berücksichtigen kann. Wie in Abschnitt 6.3.2 beschrieben, stellt die OMS sicher, dass sich die ganzzahlige ILR möglichst ähnlich zu einer konventionellen reellwertigen ILR verhält. Eine der für die technische Umsetzung besonders relevanten Eigenschaften ist der sehr flexibel anpassbare Rechenaufwand in Abhängigkeit der Größe der optimalen Menge. Damit kann die Methode an die zur Verfügung stehende Hardware angepasst werden. Somit ist das Verfahren ein sehr geeignetes Werkzeug, um eine Regelung mit ganzzahliger Eingangsbeschränkung technisch zu realisieren.

Für binäre ILR-Anwendungen konnte ebenfalls ein Ansatz vorgestellt werden, der den Rechenaufwand soweit reduziert, dass selbst für hochdimensionale Stellfolgen eine Umsetzung in Echtzeit möglich ist. Im Gegensatz zu der OMS, die sehr universell eingesetzt werden kann, ist die in Abschnitt 6.5.2 vorgestellte reduzierte ILR nur für Systeme sinnvoll anwendbar, bei denen die Stellfolge über die Zyklen deutlich mehr Nullen als Einsen enthält. Des Weiteren sind bei diesem Ansatz keine Aussagen über das Konvergenzverhalten möglich. Trotzdem zeigen die in Abschnitt 6.5 präsentierten Ergebnisse, dass in einem experimentellen Aufbau, wie dem diskutierten akustischen Plenumsversuch, mit der reduzierten ILR gute Ergebnisse erzielt werden können.

Neben den modellbasierten Ansätzen für die ganzzahlig beschränkte Regelung konnte mit dem in Kapitel 7 diskutierten EWR-Ansatz eine modellfreie Alternative vorgestellt werden. Zwar ist für diesen Ansatz ein deutlich langsames Konvergenzverhalten zu erwarten, jedoch zeigt der Vergleich mit einer ganzzahligen ILR in Abschnitt 7.2, dass im konvergierten Zustand mit beiden Ansätzen eine ähnliche Qualität im Folgeverhalten erreicht werden kann. Vor allem für sehr komplexe Systeme, wie dem in Abschnitt 7.3 beschriebenen SEC-Experiment, kann der EWR-Ansatz eine sehr effektive Alternative zu modellbasierten Ansätzen wie der ILR sein. In diesem Beispiel war die EWR in der Lage, den Zusammenhang zwischen Einspritzung und Druckanstieg im Rohr zu erlernen und so die Druckspitzen zu optimieren. Wie bei den diskutierten ILR-Methoden ist jedoch auch bei der ganzzahligen zyklischen EWR der Rechenaufwand der notwendigen Optimierung mitunter problematisch und muss bei der technischen Umsetzung berücksichtigt werden.

8.1 Ausblick

Für die in dieser Arbeit vorgestellten Methoden ergeben sich in der Zukunft die folgenden Perspektiven bzw. Herausforderungen:

- **Zustandsraummodellierung auf PINN-Basis:** Wie oben beschrieben, ist der Aufbau eines Zustandsraummodells auf PINN-Basis mit einem hohen Aufwand für die Anpassung des Trainingsprozesses verbunden. Damit der Ansatz sinnvoll und effektiv

verwendet werden kann, ist eine genauere Untersuchung des Vorgehens erforderlich, mit dem Ziel, einen einheitlichen Ablauf für die Modellidentifikation zu definieren. Der wichtigste Schritt für den Modellierungsansatz ist jedoch die Untersuchung in einem experimentellen Aufbau. Dafür müsste ein einfaches Strömungssystem aufgebaut werden, für das eine zuverlässig validierte Simulation zur Verfügung steht. Damit könnte über die Simulation eine ausreichend große Trainingsmenge generiert und ein Zustandsraummodell aufgebaut werden. Dieses Modell könnte dann in einer Zustandsschätzung oder einer modellgestützten Regelung im echten Experiment verwendet werden.

Auf der theoretischen Seite könnte eine detaillierte Untersuchung von passenden Netzstrukturen und Aktivierungsfunktionen zu einer Verbesserung der Ergebnisse führen und möglicherweise den Trainingsaufwand reduzieren. Auch der Optimierungsalgorithmus, mit dem die optimalen Parameter des Netzes ermittelt werden, hat einen größeren Einfluss auf die Trainingsergebnisse und sollte noch genauer betrachtet werden. Zusätzlich könnten komplexere Methoden, wie beispielsweise die Verwendung von sogenannte *Dropout*-Schichten, zur Verbesserung der Robustheit führen.

- **Ganzzahlige ILR:** Mit der OMS für die ganzzahlige ILR wurde eine Methode vorgestellt, die zuverlässig verwendet und deren Untersuchung als abgeschlossen angesehen werden kann. Der schwierigste Teil der Synthese ist die Lösung des in Abschnitt 6.3.2 beschriebenen nichtlinearen Optimierungsproblems. Wie bei den meisten nichtlinearen Problemen besteht auch für dieses Problem die Gefahr, in einem suboptimalen Minimum zu enden. Daher wäre es für die zukünftige Verwendung sinnvoll, das Optimierungsverfahren zu untersuchen, um möglichst gute Ergebnisse in der Synthese zu garantieren, beispielsweise durch eine passende initiale Lösung.
- **Reduzierte binäre ILR:** Die Ergebnisse in den Abschnitten 6.5.3 und 6.5.4 haben gezeigt, wie stark der Einfluss der gewählten Lösungsmenge ist. Die in dieser Arbeit gezeigten Zusammenstellungen sind nur zwei sehr einfache Möglichkeiten. Für zuverlässigere Ergebnisse muss die Zusammenstellung der Lösungsmenge weiter untersucht werden. Dabei könnte beispielsweise das Ziel im Vordergrund stehen, das Konvergenzverhalten des vollständigen Problems möglichst gut zu erhalten. Eine weitere Möglichkeit wäre eine Zusammenstellung, die möglichst viele Beschränkungen des vollständigen Problems in der reduzierten Formulierung obsolet macht. Damit könnte der Rechenaufwand für das reduzierte Problem noch weiter reduziert werden.
- **Ganzzahlige EWR:** Die Ergebnisse mit der ganzzahligen EWR, vor allem am experimentellen Aufbau der SEC, deuten auf eine breite Anwendbarkeit des Ansatzes für zyklische Prozesse hin. Für einen sicheren und zuverlässigen Betrieb sind allerdings genauere theoretische Untersuchungen des Ansatzes im Bezug auf Stabilität und Konvergenz über die Zyklen notwendig. Des Weiteren ist es vorstellbar, das sehr langsame Konvergenzverhalten zu beschleunigen, indem nicht alle Einträge der Stellfolge in jedem Optimierungsschritt perturbiert werden. Allerdings ist davon auszugehen, dass dadurch möglicherweise die Robustheit, beispielsweise im Bezug auf Rauscheinflüsse, verschlechtert wird.

9 Anhang

A Finite-Volumen-Methode

Dieser Abschnitt stellt kurz die grundlegende Idee der Finite-Volumen-Methode (FVM) vor. Die Erklärung orientiert sich dabei an den Ausführungen in Berndt, 2016. Theoretische Details der FVM sind in LeVeque, 2002 und Toro, 2013 zu finden. Da sich die Beispiele dieser Arbeit alle auf den eindimensionalen Fall beschränken, wird auch hier von einem System ausgegangen, das nur in einer räumlichen Richtung x beschrieben wird. Ausgangspunkt ist eine partielle Differentialgleichung

$$\frac{\partial}{\partial t} h(x, t) + \frac{\partial}{\partial x} f_h(h(x, t)) = 0 \quad (\text{A.1})$$

in der Erhaltungsform. Hier wird von einer eindimensionalen Lösung $h(x, t)$ der partiellen Differentialgleichung ausgegangen. Die weiteren Schritte gelten allerdings analog für eine vektorwertige Beschreibung eines Gleichungssystems wie das der Euler-Gleichungen. Die Funktion f_h ist eine beliebige Abbildung $\mathbb{R} \rightarrow \mathbb{R}$ in Abhängigkeit von h . Für einen Raum-Zeit-Bereich mit $x \in [x_0, x_1]$ und $\tau \in [t, t + \Delta t]$ kann die partielle Differentialgleichung in die integrale Form

$$\int_{x_0}^{x_1} h(x, \tau) dx \Big|_{\tau=t}^{\tau=t+\Delta t} + \int_t^{t+\Delta t} f_h(h(x, \tau)) d\tau \Big|_{x=x_0}^{x=x_1} = 0 \quad (\text{A.2})$$

gebracht werden. Wird die räumliche Dimension in n_Z Intervalle

$$[x_0, x_1], \dots, [x_{i-1}, x_i], \dots, [x_{n_Z-1}, x_{n_Z}] \quad (\text{A.3})$$

unterteilt, kann für das i -te Intervall eine räumliche Mittelung

$$\bar{h}_i(\tau) = \frac{1}{x_i - x_{i-1}} \int_{x_{i-1}}^{x_i} h(x, \tau) dx \quad (\text{A.4})$$

für die Erhaltungsgröße bestimmt werden. Die integrale Form aus Gl. (A.2) kann genutzt werden, um für das i -te räumliche Intervall den Zusammenhang

$$\int_{x_{i-1}}^{x_i} h(x, t + \Delta t) dx = \int_{x_{i-1}}^{x_i} h(x, t) dx - \int_t^{t+\Delta t} f_h(h(x, \tau)) d\tau \Big|_{x=x_{i-1}}^{x=x_i} \quad (\text{A.5})$$

zwischen den Zeitpunkten t und $t + \Delta t$ herzustellen. Wird nun die Mittelung aus Gl. (A.4) in Gl. (A.5) eingesetzt, ergibt sich der Zeitschritt

$$\bar{h}_i(t + \Delta t) = \bar{h}_i(t) - \frac{1}{x_i - x_{i-1}} \left(\int_t^{t+\Delta t} f_h(h(x_i, \tau)) d\tau - \int_t^{t+\Delta t} f_h(h(x_{i-1}, \tau)) d\tau \right), \quad (\text{A.6})$$

der das numerische Verfahren der FVM definiert.

Damit der Ansatz aus Gl. (A.6) verwendet werden kann, müssen die Werte an den Intervallgrenzen x_{i-1} und x_i bestimmt werden. Dafür ist jeweils ein auf die partielle Differentialgleichung aus Gl. (A.1) bezogenes Anfangswertproblem mit

$$h(x, 0) = \begin{cases} \bar{h}_i(0) & \text{für } x < x_i \\ \bar{h}_{i+1}(0) & \text{für } x > x_i \end{cases} \quad (\text{A.7})$$

an den entsprechenden Stellen $x = x_{i-1}$ und $x = x_i$ zu lösen. Auf die Lösung dieses sogenannten Riemann-Problems wird an dieser Stelle nicht weiter eingegangen, sondern auf die Ausführungen in Berndt, 2016 verwiesen. Bei ausreichend genauer Lösung des Riemann-Problems und passender Wahl der Zeitschrittweite Δt kann die Courant-Friedrichs-Lewy-Bedingung (CFL-Bedingung) für die FVM erfüllt werden und numerische Stabilität ist sichergestellt.

Das beschriebene Verfahren ist exakt in der Zeit. Bei der räumlichen Genauigkeit gilt aber nur eine Approximationsgenauigkeit erster Ordnung. Eine verbesserte räumliche Genauigkeit kann jedoch erreicht werden, wenn die Werte $\bar{h}_i(0)$ und $\bar{h}_{i+1}(0)$ aus dem Riemann-Problem in Gl. (A.7) über eine Interpolation aus mehreren benachbarten Zellen bestimmt werden.

B BFGS-Algorithmus

Die Broyden–Fletcher–Goldfarb–Shanno-Methode (BFGS-Methode) ist ein häufig genutztes Quasi-Newton-Verfahren. Der Ansatz kann genutzt werden, um eine skalare Funktion

$$J(\boldsymbol{\mu}) \quad (\text{B.1})$$

in Abhängigkeit eines Vektors $\boldsymbol{\mu}$ iterativ zu minimieren. Dieser Abschnitt erklärt kurz die grundlegende Idee der BFGS-Methode. Für die Ausführungen wird sich dabei an der Erklärung in Nocedal und Wright, 2006 orientiert. Weitere Details zum Konvergenzverhalten der Methode sind in Dai, 2003 zu finden. Um die Notation im Folgenden zu vereinfachen, werden Werte, die sich in Abhängigkeit von $\boldsymbol{\mu}_k$ im Zyklus k ergeben, mit einem entsprechenden Index gekennzeichnet. Es gilt also $J(\boldsymbol{\mu}_k) = J_k$.

Ausgangspunkt ist eine quadratische Approximation der Gl. (B.1)

$$\hat{J}_k(\mathbf{h}) = J_k + \nabla J_k^T \mathbf{h} + \frac{1}{2} \mathbf{h}^T \hat{\mathbf{H}}_k \mathbf{h} \quad (\text{B.2})$$

in der Iteration k . Die Matrix $\hat{\mathbf{H}}$ entspricht an dieser Stelle einer Schätzung der Hesse-Matrix. Diese Approximation wird minimal für

$$\mathbf{h}_k = -\hat{\mathbf{H}}_k^{-1} \nabla J_k \quad (\text{B.3})$$

Damit kann in Abhängigkeit einer Schrittweite ζ_k ein neuer Wert

$$\boldsymbol{\mu}_{k+1} = \boldsymbol{\mu}_k + \zeta_k \mathbf{h}_k \quad (\text{B.4})$$

für die nächste Iteration bestimmt werden. Der Ansatz der BFGS-Methode ist eine iterative Anpassung der Näherung der Hesse-Matrix in Abhängigkeit der vorangegangenen Iteration anstelle einer immer wieder neu berechneten numerischen Approximation in jeder Iteration.

Für die nächste Iteration $k+1$ kann ebenfalls eine quadratische Approximation der Funktion aus Gl. (B.1)

$$\hat{J}_{k+1}(\mathbf{h}) = J_{k+1} + \nabla J_{k+1}^T \mathbf{h} + \frac{1}{2} \mathbf{h}^T \hat{\mathbf{H}}_{k+1} \mathbf{h} \quad (\text{B.5})$$

angenommen werden. Die Schätzung der Hesse-Matrix soll nun so gewählt werden, dass für den Gradienten der Approximation

$$\nabla \hat{J}_{k+1}(\mathbf{0}) = \nabla J_{k+1} \quad (\text{B.6})$$

$$\nabla \hat{J}_{k+1}(-\zeta_k \mathbf{h}_k) = \nabla J_k \quad (\text{B.7})$$

gilt. Damit ist sichergestellt, dass an den Stellen $\boldsymbol{\mu}_k$ und $\boldsymbol{\mu}_{k+1}$ der Gradient der Approximation dem realen Gradienten entspricht. Die Bedingung aus Gl. (B.6) ist immer automatisch erfüllt. Um die Bedingung aus Gl. (B.7) zu erfüllen, muss gemäß Gl. (B.5)

$$\nabla J_{k+1} - \hat{\mathbf{H}}_{k+1} \zeta_k \mathbf{h}_k = \nabla J_k \quad (\text{B.8})$$

$$\Leftrightarrow \hat{\mathbf{H}}_{k+1} \zeta_k \mathbf{h}_k = \nabla J_{k+1} - \nabla J_k \quad (\text{B.9})$$

gelten. Die Bedingung wird häufig als Sekantengleichung bezeichnet. Für eine einfachere Lesbarkeit wird im Folgenden die Notation

$$\Delta \boldsymbol{\mu}_k = \boldsymbol{\mu}_{k+1} - \boldsymbol{\mu}_k = \zeta_k \mathbf{h}_k \quad (\text{B.10})$$

$$\mathbf{g}_k = \nabla J_{k+1} - \nabla J_k \quad (\text{B.11})$$

verwendet. Ausgehend von der Annahme, dass die Schätzung der Hesse-Matrix symmetrisch und positiv definit ist, muss entsprechend

$$\Delta\boldsymbol{\mu}_k^T \mathbf{g}_k > 0 \quad (\text{B.12})$$

gelten, damit die Sekantengleichung erfüllt werden kann. Für konvexe Probleme ist dies immer erfüllt. Für den nichtkonvexen Fall und weiter Details sei hier auf die Erklärungen in Nocedal und Wright, 2006 verwiesen.

Da es unendlich viele Lösungen der Sekantengleichung gibt, muss eine zusätzliche Beschränkung eingeführt werden. Eine sinnvolle Forderung ist beispielsweise eine möglichst kleine Änderung der geschätzten Hesse-Matrix von einer Iteration zur nächsten. Als Maß für die Änderung kann eine beliebige Norm gewählt werden. Die Wahl der Norm bestimmt, welches Quasi-Newton-Verfahren sich ergibt. Wird die Frobenius-Matrixnorm $\|\cdot\|_F$ gewählt, ergibt sich das sogenannte Davidon-Fletcher-Powell-Verfahren (DFP-Verfahren), welches als Teilschritt der eigentlichen BFGS-Methode zu verstehen ist. Die Hesse-Matrix ist in diesem Fall über

$$\hat{\mathbf{H}}_{k+1} = \arg \min_{\hat{\mathbf{H}}} \left\| \hat{\mathbf{H}} - \hat{\mathbf{H}}_k \right\|_F \quad (\text{B.13})$$

unter der Beschränkung

$$\hat{\mathbf{H}} = \hat{\mathbf{H}}^T \quad \text{und} \quad \hat{\mathbf{H}} \Delta\boldsymbol{\mu}_k = \mathbf{g}_k \quad (\text{B.14})$$

definiert. Die Details der Lösung dieses Optimierungsproblem ist in Nocedal und Wright, 2006 zu finden. Es ergibt sich die Anpassungsregel für die geschätzte Hesse-Matrix

$$\hat{\mathbf{H}}_{k+1} = \left(\mathbf{I} - \frac{\mathbf{g}_k \Delta\boldsymbol{\mu}_k^T}{\mathbf{g}_k^T \Delta\boldsymbol{\mu}_k} \right) \hat{\mathbf{H}}_k \left(\mathbf{I} - \frac{\Delta\boldsymbol{\mu}_k \mathbf{g}_k^T}{\mathbf{g}_k^T \Delta\boldsymbol{\mu}_k} \right) + \frac{\mathbf{g}_k \mathbf{g}_k^T}{\mathbf{g}_k^T \Delta\boldsymbol{\mu}_k} \quad (\text{B.15})$$

von der Iteration k zur Iteration $k+1$. Der Ablauf einer Minimierung mit der DFP-Methode ist in Algorithmus 3 dargestellt.

Algorithmus 3 DFP/BFGS-Algorithmus

```

1:  $\hat{\mathbf{H}}_k = \hat{\mathbf{H}}_0$  ▷ Initialisierung der geschätzten Hesse-Matrix
2:  $\epsilon = \epsilon_0$  ▷ Initialisierung der Konvergenzgrenze
3:  $\boldsymbol{\mu}_k = \boldsymbol{\mu}_0$  ▷ Startpunkt der Suche
4:  $\nabla J_k = \text{GRADIENT}(J, \boldsymbol{\mu}_k)$ 
5: while  $\|\nabla J_k\|_2 > \epsilon$  do
6:    $\mathbf{h}_k = -\hat{\mathbf{H}}_k^{-1} \nabla J_k$ 
7:    $\zeta_k = \text{DETERMINESTEPsize}()$  ▷ Schrittweite bestimmen (z.B. Linesearch)
8:    $\boldsymbol{\mu}_{k+1} = \boldsymbol{\mu}_k + \zeta_k \mathbf{h}_k$ 
9:    $\nabla J_{k+1} = \text{GRADIENT}(J, \boldsymbol{\mu}_{k+1})$ 
10:   $\Delta\boldsymbol{\mu}_k = \boldsymbol{\mu}_{k+1} - \boldsymbol{\mu}_k$ 
11:   $\mathbf{g}_k = \nabla J_{k+1} - \nabla J_k$ 
12:   $\hat{\mathbf{H}}_{k+1} = \text{UPDATE}(\Delta\boldsymbol{\mu}_k, \mathbf{g}_k, \hat{\mathbf{H}}_k)$ 
13:   $\boldsymbol{\mu}_k = \boldsymbol{\mu}_{k+1}$ 
14:   $\nabla J_k = \text{GRADIENT}(J, \boldsymbol{\mu}_k)$ 
15: end while

```

Da in diesem Algorithmus ausschließlich die Inverse der Hesse-Matrix verwendet wird, kann auch direkt die inverse Form iterativ berechnet werden. Diese kann analog zu Gl. (B.13) über

$$\hat{\mathbf{H}}_{k+1}^{-1} = \arg \min_{\hat{\mathbf{H}}^{-1}} \left\| \hat{\mathbf{H}}^{-1} - \hat{\mathbf{H}}_k^{-1} \right\|_{\text{F}} \quad (\text{B.16})$$

unter der Beschränkung

$$\hat{\mathbf{H}}^{-1} = \left(\hat{\mathbf{H}}^{-1} \right)^{\text{T}} \quad \text{und} \quad \hat{\mathbf{H}}^{-1} \mathbf{g}_k = \Delta \boldsymbol{\mu}_k \quad (\text{B.17})$$

bestimmt werden. Damit ergibt sich die Anpassungsregel

$$\hat{\mathbf{H}}_{k+1}^{-1} = \left(\mathbf{I} - \frac{\Delta \boldsymbol{\mu}_k \mathbf{g}_k^{\text{T}}}{\mathbf{g}_k^{\text{T}} \Delta \boldsymbol{\mu}_k} \right) \hat{\mathbf{H}}_k^{-1} \left(\mathbf{I} - \frac{\mathbf{g}_k \Delta \boldsymbol{\mu}_k^{\text{T}}}{\mathbf{g}_k^{\text{T}} \Delta \boldsymbol{\mu}_k} \right) + \frac{\Delta \boldsymbol{\mu}_k \Delta \boldsymbol{\mu}_k^{\text{T}}}{\mathbf{g}_k^{\text{T}} \Delta \boldsymbol{\mu}_k} \quad (\text{B.18})$$

für die inverse Hesse-Matrix. Im engeren Sinne wird diese Anpassung als BFGS-Methode bezeichnet und kann in Zeile 11 von Algorithmus 3 genutzt werden, um direkt die Schätzung der inversen Hesse-Matrix zu berechnen.

C Branch-and-Bound-Suche für ganzzahlige quadratische Programme

Die Grundlage für die meisten ganzzahligen bzw. gemischt ganzzahligen QP-Löser ist die sogenannte *Branch-and-Bound*-Suche. Auch die gängigen kommerziellen Löser, *Gurobi* (Gurobi Optimization, 2020) und *CPLEX* (Blick et al., 2014), basieren grundlegend auf diesem Verfahren. Ein Überblick über die *Branch-and-Bound*-Methode kann in Morrison et al., 2016 gefunden werden.

Durch den diskreten Charakter des Lösungsraums des QPs ist es möglich, durch eine Suche die möglichen Lösungen Stück für Stück zu explorieren. Ausgangspunkt ist ein allgemeines, reellwertiges QP.

Quadratisches Programm (QP) 11: *Die reellwertige Lösung eines durch die Parameter \mathbf{Q} und \mathbf{q} definierten, quadratischen Minimierungsproblem ergibt sich aus*

$$\boldsymbol{\mu} = \arg \min_{\tilde{\boldsymbol{\mu}}} \tilde{\boldsymbol{\mu}}^T \mathbf{Q} \tilde{\boldsymbol{\mu}} + \mathbf{q}^T \tilde{\boldsymbol{\mu}} \quad (\text{C.1})$$

mit der oberen und unteren Schranke

$$\mathbf{c}_{\text{lb}} \leq \tilde{\boldsymbol{\mu}} \leq \mathbf{c}_{\text{ub}} \quad . \quad (\text{C.2})$$

Zusätzliche Beschränkungen werden im Folgenden vernachlässigt, können aber in der Lösung von QP 11 problemlos berücksichtigt werden. Für einen Explorationsschritt auf der Suche nach der optimalen ganzzahligen Lösung wird nun zunächst die reellwertige Lösung $\boldsymbol{\mu} = [\mu_1, \dots, \mu_i, \dots, \mu_n]$ von QP 11 bestimmt. Diese Lösung ist im Allgemeinen nicht ganzzahlig. Daher werden zwei neue QPs aufgestellt. Dafür wird eine Variable μ_i der Lösung ausgewählt, die nicht ganzzahlig ist. Mit den Rundungsoperatoren $\lfloor \cdot \rfloor$ und $\lceil \cdot \rceil$ für Abrunden bzw. Aufrunden können zwei neue Optimierungsprobleme definiert werden.

Quadratisches Programm (QP) 12: *Die optimale Lösung des linken Zweiges eines Explorationsschritts zur Bestimmung der ganzzahligen Lösung von QP 11 ergibt sich aus*

$$\boldsymbol{\mu} = \arg \min_{\tilde{\boldsymbol{\mu}}} \tilde{\boldsymbol{\mu}}^T \mathbf{Q} \tilde{\boldsymbol{\mu}} + \mathbf{q}^T \tilde{\boldsymbol{\mu}} \quad (\text{C.3})$$

mit der oberen und unteren Schranke

$$\mathbf{c}_{\text{lb}}^1 \leq \tilde{\boldsymbol{\mu}} \leq \mathbf{c}_{\text{ub}}^1 \quad \mathbf{c}_{\text{lb}}^1 = \mathbf{c}_{\text{lb}} \quad \mathbf{c}_{\text{ub}}^1 = [c_{\text{ub},1}, \dots, \lfloor \mu_i \rfloor, \dots, c_{\text{ub},n}] \quad . \quad (\text{C.4})$$

Quadratisches Programm (QP) 13: *Die optimale Lösung des rechten Zweiges eines Explorationsschritts zur Bestimmung der ganzzahligen Lösung von QP 11 ergibt sich aus*

$$\boldsymbol{\mu} = \arg \min_{\tilde{\boldsymbol{\mu}}} \tilde{\boldsymbol{\mu}}^T \mathbf{Q} \tilde{\boldsymbol{\mu}} + \mathbf{q}^T \tilde{\boldsymbol{\mu}} \quad (\text{C.5})$$

mit der oberen und unteren Schranke

$$\mathbf{c}_{\text{lb}}^2 \leq \tilde{\boldsymbol{\mu}} \leq \mathbf{c}_{\text{ub}}^2 \quad \mathbf{c}_{\text{lb}}^2 = [c_{\text{lb},1}, \dots, \lceil \mu_i \rceil, \dots, c_{\text{lb},n}] \quad \mathbf{c}_{\text{ub}}^2 = \mathbf{c}_{\text{ub}} \quad . \quad (\text{C.6})$$

Für die beiden QPs kann dann wiederum jeweils ein neuer Explorationsschritt durchgeführt werden. Eine Visualisierung des Explorationsschritts als Ausschnitt eines Suchbaums ist in Abb. 9.1 gezeigt.

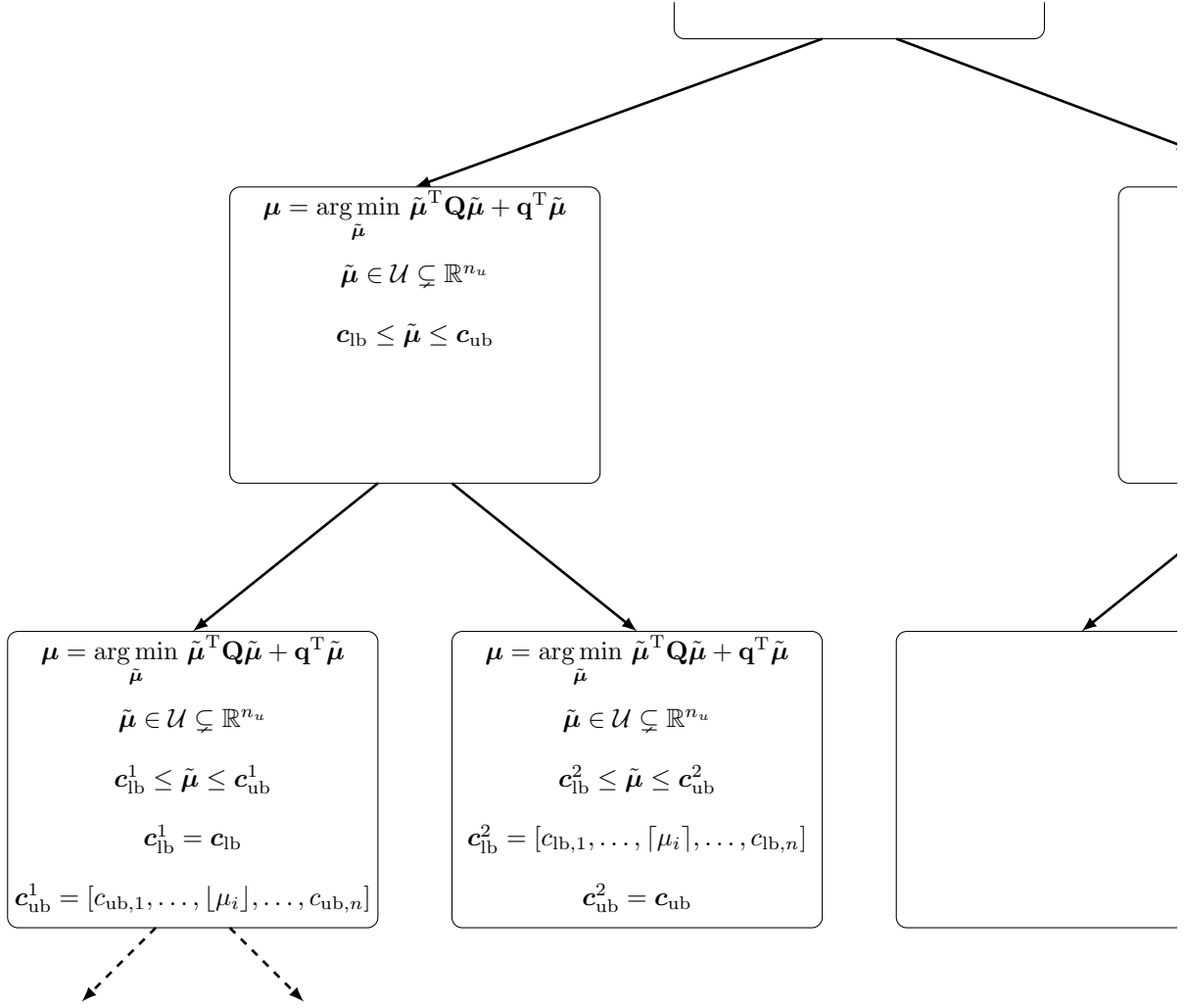


Abbildung 9.1: Visualisierung eines Branch-and-Bound-Suchbaums zur Lösung eines ganzzahligen QPs

Während der Exploration bilden sich einzelne Zweige, deren Blätter ganzzahligen Lösungen bzw. nicht zulässigen Lösungen bezüglich den Beschränkungen entsprechen. Um sicherzustellen, dass nur aussichtsreiche Zweige bis zum Ende exploriert werden, kann in jedem Schritt nicht nur die Lösung μ ausgewertet werden, sondern auch der entsprechende Kostenwert

$$J = \mu^T Q \mu + \mathbf{q}^T \mu \quad . \quad (\text{C.7})$$

Ist dieser Wert größer als der Kostenwert der bisher besten zulässigen Lösung, wird der entsprechende Zweig nicht mehr weiter exploriert. Der Algorithmus terminiert, wenn keine Knoten mehr zur Exploration zur Verfügung stehen. Der genaue Ablauf der Optimierung ist Algorithmus 4 zu entnehmen.

Optimierungen

Auch wenn die kommerziellen, ganzzahligen QP-Löser auf den gleichen Algorithmus aufbauen, gibt es verschiedenste Optimierungsmöglichkeiten, die in den einzelnen Software-Paketeten abweichend umgesetzt sein können. Die genauen Details der Umsetzung einzelner Ansätze

Algorithmus 4 Grundlegender *Branch-and-Bound*-Algorithmus

```

1:  $QP_0 = \text{CREATEQP}(c_{lb}, c_{ub}, \mathbf{Q}, \mathbf{q}, \mathcal{U})$ 
2:  $QPs = \{ QP_0 \}$  ▷ noch zu lösende QPs
3:  $S = \{ \}$  ▷ Menge aller gefundenen zulässiger Lösungen
4: while  $\text{ISEMPTY}(QPs) == \text{FALSE}$  do
5:    $QP = \text{PICKQP}(QPs)$  ▷ QP auswählen, das als Nächstes gelöst wird
6:    $J, \mu = \text{SOLVEQP}(QP)$ 
7:   if  $\text{ISEMPTY}(u) == \text{FALSE}$  and  $J < \text{SELECTBESTCOST}(S)$  then
8:     if  $\text{ISINTEGER}(u) == \text{TRUE}$  then
9:        $S = \text{ADDSOLUTION}(S, (J, \mu))$ 
10:    else
11:       $\mu_i = \text{SELECTSPLIT}(\mu)$  ▷ Auswahl der Variable für den Branch
12:       $c_{lb} = \text{GETLOWERBOUND}(QP)$ 
13:       $c_{ub} = \text{GETUPPERBOUND}(QP)$ 
14:       $c_{lb}^1 = c_{lb}$ 
15:       $c_{ub}^1 = \text{MERGEBOUND}(c_{ub}, \mu_i)$  ▷ Neue Grenzen gemäß Gl. (C.4)
16:       $c_{lb}^2 = \text{MERGEBOUND}(c_{lb}, \mu_i)$  ▷ Neue Grenzen gemäß Gl. (C.6)
17:       $c_{ub}^2 = c_{ub}$ 
18:       $QP1 = \text{CREATEQP}(c_{lb}^1, c_{ub}^1, \mathbf{Q}, \mathbf{q}, \mathcal{U})$ 
19:       $QP2 = \text{CREATEQP}(c_{lb}^2, c_{ub}^2, \mathbf{Q}, \mathbf{q}, \mathcal{U})$ 
20:       $QPs = \text{ADDQP}(QP1, QPs)$  ▷ QP1 in Liste der zu lösenden QPs einfügen
21:       $QPs = \text{ADDQP}(QP2, QPs)$  ▷ QP2 in Liste der zu lösenden QPs einfügen
22:    end if
23:  end if
24: end while
25: return  $\text{SELECTBESTSOLUTION}(S)$ 

```

sind üblicherweise Unternehmensgeheimnis. Um ein besseres Verständnis der Algorithmen zu ermöglichen, soll hier trotzdem kurz auf verbreitete Ansätze eingegangen werden.

Der Algorithmus 4 bestimmt zwar den allgemeinen Rahmen der Exploration, trotzdem sind bestimmte Schritte nicht genau spezifiziert. Diese Entwurfsentscheidungen werden als Heuristiken zusammengefasst. Die gängigsten Ansätze sind:

- **Initiale Lösungen** (Zeile 3): In manchen Fällen ist es möglich, bereits vor der eigentlichen Exploration eine oder mehrere zulässige Lösung zu erraten. Dadurch können bestimmte Zweige des Suchbaums frühzeitig ausgeschlossen werden.
- **Auswahl des zu lösenden QP** (Zeile 5): Je nach Auswahl, welches QP als nächstes ausgewertet wird, kann die Reihenfolge der Exploration des Suchbaums beeinflusst werden. Dafür können z.B. die Kosten vorangegangener Explorationsschritte in dem entsprechenden Zweig berücksichtigt werden oder die Beschränkungen. Bei der Definition der Auswahlstrategie muss allerdings berücksichtigt werden, dass ein komplizierter Auswahlalgorithmus möglicherweise sehr rechenaufwendig ist. Damit kann möglicherweise die erhoffte Verbesserung in der Performance wieder negiert wird. Dies ist vor allem für sehr große Optimierungsprobleme relevant, bei denen die Liste der noch auszuwertenden QPs sehr groß wird. Häufig wird daher einfach das erste QP in der Liste ausgewählt.
- **Einfügen neuer QPs** (Zeile 20, 21): Ebenso wie die Auswahl aus der Liste hat auch das

Einfügen der QPs in die Liste Einfluss auf die Explorationsreihenfolge des Suchbaums. Die beiden gängigsten Strategien sind ausgehend davon, dass immer das erste QP aus der Liste für die Exploration ausgewählt wird, die Liste mit einer *first-in-first-out*- oder *first-in-last-out*-Strategie zu verwenden. Dadurch wird entweder eine Tiefen- oder Breitensuche realisiert. Auch ein Wechsel zwischen verschiedenen Einfügestrategien im Verlauf der Suche ist möglich. So kann es sinnvoll sein, zunächst mit einer Tiefensuche das schnelle Finden einer zulässigen Lösung zu forcieren und im Anschluss zu einer Breitensuche zu wechseln.

- **Wahl des Branches** (Zeile 11): Wenn die Lösung eines QPs aus mehreren nicht ganzzahligen Werten besteht, muss entschieden werden, für welche Variable der nächste Branch durchgeführt wird. Eine mögliche Regel ist die Variable zu wählen, deren Wert am weitesten von einer ganzzahligen Lösung entfernt ist. Ausgehend von der Annahme einer ähnlichen Sensitivität aller Variablen werden damit Explorationsschritte mit starken Veränderungen in der Kostenfunktion in den oberen Teil des Suchbaums verlegt.

Literatur

- Ahmed, N. K., Atiya, A. F., El Gayar, N. & El-Shishiny, H. (2010). An Empirical Comparison of Machine Learning Models for Time Series Forecasting. *Econometric Reviews*, 29(5), 594–621. doi:10.1080/07474938.2010.481556
- Ahn, H. S., Chen, Y. Q. & Moore, K. L. (2007). Iterative Learning Control: Brief Survey and Categorization. *IEEE Transactions on Systems, Man and Cybernetics Part C: Applications and Reviews*, 37(6), 1099–1121. doi:10.1109/TSMCC.2007.905759
- Aleksić, K., King, R., Noack, B. R., Lehmann, O., Morzynski, M. & Tadmor, G. (2008). Nonlinear Flow Control Using a Low Dimensional Galerkin Model. *Automatic Control and Robotics*, 7(1), 63–70.
- Amann, N., Owens, D. H. & Rogers, E. (1998). Predictive optimal iterative learning control. *International Journal of Control*, 69(2), 203–226. doi:10.1080/002071798222794
- Andino, M. Y., Lin, J. C., Roman, S., Graff, E. C., Gharib, M., Whalen, E. A. & Wygnanski, I. J. (2019). Active Flow Control on Vertical Tail Models. *AIAA Journal*, 57(8), 3322–3328. doi:10.2514/1.J057876
- Ansell, P. J. & Haran, K. S. (2020). Electrified Airplanes: A Path to Zero-Emission Air Travel. *IEEE Electrification Magazine*, 8(2), 18–26. doi:10.1109/MELE.2020.2985482
- Apicella, A., Donnarumma, F., Isgrò, F. & Prevete, R. (2021). A survey on modern trainable activation functions. *Neural Networks*, 138. doi:10.1016/j.neunet.2021.01.026
- Ariyur, K. B. & Krstic, M. (2003). *Real-time optimization by extremum-seeking control*. John Wiley & Sons.
- Arnold, F. & King, R. (2021a). Norm-optimal iterative learning control in an integer-valued control domain. *International Journal of Control*. doi:10.1080/00207179.2021.1983213
- Arnold, F. & King, R. (2021b). State-space modeling for control based on physics-informed neural networks. *Engineering Applications of Artificial Intelligence*, 101, 104195. doi:10.1016/j.engappai.2021.104195
- Arnold, F., King, R., Yücel, F., Völzke, F. & Paschereit, C. O. (2019). Modelling of Fuel Transport for a Shockless Explosion Combustion Process. In *AIAA Scitech 2019 Forum* (S. 1254). doi:10.2514/6.2019-1254
- Arnold, F., Neuhäuser, K. & King, R. (2020). Experimental Comparison of Two Integer Valued Iterative Learning Control Approaches at a Stator Cascade. *Journal of Engineering for Gas Turbines and Power*, 142(1), 011002. doi:10.1115/1.4045339
- Arnold, F., Topalovic, D. & King, R. (2020). Efficient Implementation of a Binary Iterative Learning Control. *IFAC-PapersOnLine*, 53(2), 1499–1504. doi:10.1016/j.ifacol.2020.12.2007
- Aubrun, S., Leroy, A. & Devinant, P. (2017). A review of wind turbine-oriented active flow control strategies. *Experiments in Fluids*, 58(134). doi:10.1007/s00348-017-2412-0
- Axehill, D., Vandenbergh, L. & Hansson, A. (2010). Convex relaxations for mixed integer predictive control. *Automatica*, 46(9), 1540–1545. doi:10.1016/j.automatica.2010.06.015
- Badran, O. O. (1999). Gas-turbine performance improvements. *Applied Energy*, 64(1-4), 263–273. doi:10.1016/S0306-2619(99)00088-4

- Balsara, D. S. (2010). Multidimensional HLLC Riemann solver: Application to Euler and magnetohydrodynamic flows. *Journal of Computational Physics*, 229(6), 1970–1993. doi:10.1016/j.jcp.2009.11.018
- Bansal, H., Schulze, P., Abbasi, M. H., Zwart, H., Iapichino, L., Schilders, W. H. A. & van de Wouw, N. (2021). Port-Hamiltonian formulation of two-phase flow models. *Systems & Control Letters*, 149, 104881. doi:10.1016/j.sysconle.2021.104881
- Baydin, A. G., Pearlmutter, B. A., Radul, A. A. & Siskind, J. M. (2018). Automatic Differentiation in Machine Learning: A Survey. *Journal of Machine Learning Research*, 18, 1–43. Zugriff unter <https://www.jmlr.org/papers/volume18/17-468/17-468.pdf>
- Bemporad, A. & Morari, M. (1999). Control of systems integrating logic, dynamics, and constraints. *Automatica*, 35(3), 407–427. doi:10.1016/S0005-1098(98)00178-2
- Berkooz, G., Holmes, P. & Lumley, J. L. (1993). The Proper Orthogonal Decomposition in the Analysis of Turbulent Flows. *Annual Review of Fluid Mechanics*, 25(1), 539–575. doi:10.1146/annurev.fl.25.010193.002543
- Berndt, P. (2014). On the Use of the HLL-Scheme for the Simulation of the Multi-Species Euler Equations. *Springer Proceedings in Mathematics & Statistics*, (78), 809–816. doi:10.1007/978-3-319-05591-6_81
- Berndt, P. (2016). *Mathematical Modeling of the Shockless Explosion Combustion* (Diss., Freie Universität Berlin).
- Berndt, P. & Klein, R. (2016). Modeling the kinetics of the Shockless Explosion Combustion. *Combustion and Flame*, 175, 16–26. doi:10.1016/j.combustflame.2016.06.029
- Blik, C., Bonami, P. & Lodi, A. (2014). Solving Mixed-Integer Quadratic Programming problems with IBM-CPLEX: a progress report. *Proceedings of the Twenty-Sixth RAMP Symposium Hosei University, Tokyo*, 171–180. Zugriff unter <http://www.orsj.or.jp/ramp/2014/paper/4-3.pdf>
- Bobusch, B. C., Berndt, P., Paschereit, C. O. & Klein, R. (2014). Shockless Explosion Combustion: An Innovative Way of Efficient Constant Volume Combustion in Gas Turbines. *Combustion Science and Technology*, 186(10-11), 1680–1689. doi:10.1080/00102202.2014.935624
- Bristow, D. A., Tharayil, M. & Alleyne, A. G. (2006). A survey of iterative learning control. *IEEE Control Systems Magazine*, 26(June), 96–114. doi:10.1109/MCS.2006.1636313
- Brunton, S. L., Noack, B. R. & Koumoutsakos, P. (2020). Machine Learning for Fluid Mechanics. *Annual Review of Fluid Mechanics*, 52(1), 477–508. doi:10.1146/annurev-fluid-010719-060214
- Buchheim, C., Santis, M. D. E., Lucidi, S., Rinaldi, F. & Trieu, L. (2016). A Feasible Active Set Method with Reoptimization for Convex Quadratic Mixed-Integer Programming. *SIAM Journal on Optimization*, 26(3), 1695–1714. doi:10.1137/140978971
- Bürger, A., Zeile, C., Altmann-Dieses, A., Sager, S. & Diehl, M. (2019). Design, implementation and simulation of an MPC algorithm for switched nonlinear systems under combinatorial constraints. *Journal of Process Control*, 81, 15–30. doi:10.1016/j.jprocont.2019.05.016
- Bussig, T. & Pappas, G. (1996). Pulse Detonation Engine Theory And Concepts. *AIAA Developments In High-Speed Vehicle Propulsion Systems*, 421–472. doi:10.2514/5.9781600866401.0421.0472
- Butcher, J. C. (1996). A history of Runge-Kutta methods. *Applied Numerical Mathematics*, 20, 247–260. doi:10.1016/0168-9274(95)00108-5
- Byrd, R. H., Hribar, M. E. & Nocedal, J. (1999). An Interior Point Algorithm for Large-Scale Nonlinear Programming. *SIAM Journal on Optimization*, 9(4), 877–900. doi:10.1137/S1052623497325107

- Caldwell, N., Glaser, A., Dimicco, R. & Gutmark, E. (2005). Acoustic Measurements of an Integrated Pulse Detonation Engine with Gas Turbine System. In *43rd AIAA Aerospace Sciences Meeting and Exhibit* (S. 413). doi:10.2514/6.2005-413
- Cattafesta, L. N. & Sheplak, M. (2011). Actuators for Active Flow Control. *Annual Review of Fluid Mechanics*, 43, 247–272. doi:10.1146/annurev-fluid-122109-160634
- Choi, B. (2012). *ARMA Model Identification*. Springer Science & Business Media. doi:10.1007/978-1-4613-9745-8
- Chu, B. & Owens, D. H. (2010). Iterative learning control for constrained linear systems. *International Journal of Control*, 83(7), 1397–1413. doi:10.1080/00207171003758752
- Cimini, G. & Bemporad, A. (2017). Exact Complexity Certification of Active-Set Methods for Quadratic Programming. *IEEE Transactions on Automatic Control*, 62(12), 6094–6109. doi:10.1109/tac.2017.2696742
- Collis, S. S., Joslin, R. D., Seifert, A. & Theofilis, V. (2004). Issues in active flow control: Theory, control, simulation, and experiment. *Progress in Aerospace Sciences*, 40(4-5), 237–289. doi:10.1016/j.paerosci.2004.06.001
- Constantin, P. & Foias, C. (1988). *Navier-Stokes equations*. University of Chicago Press.
- Dai, Y. (2003). Convergence Properties of the BFGS Algorithm. *SIAM Journal on Optimization*, 13(3), 693–701. doi:10.1137/S1052623401383455
- Dakin, R. J. (1965). A tree-search algorithm for mixed integer programming problems. *The Computer Journal*, 8(3), 250–255. doi:10.1093/comjnl/8.3.250
- Desai, M. & Ito, K. (1994). Optimal Controls of Navier-Stokes Equations. *SIAM Journal on Control and Optimization*, 32(5), 1428–1446. doi:10.1137/S0363012992224972
- Dhert, T., Ashuri, T. & Martins, J. R. R. A. (2017). Aerodynamic shape optimization of wind turbine blades using a Reynolds-averaged Navier–Stokes model and an adjoint method. *Wind Energy*, 20(5), 909–926. doi:10.1002/we.2070
- Duriez, T., Brunton, S. L. & Noack, B. R. (2017). *Machine Learning Control – Taming Nonlinear Dynamics and Turbulence* (1. Aufl.). Springer International Publishing.
- Einfeldt, B. (1988). On Godunov-Type Methods for Gas Dynamics. *SIAM Journal on Numerical Analysis*, 25(2), 294–318. doi:10.1137/0725021
- Endo, T., Kasahara, J., Matsuo, A., Inaba, K., Sato, S. & Fujiwara, T. (2004). Pressure History at the Thrust Wall of a Simplified Pulse Detonation Engine. *AIAA Journal*, 42(9), 1921–1930. doi:10.2514/1.976
- Eymard, R., Gallouet, T. & Herbin, R. (2000). Finite volume methods. *Handbook of Numerical Analysis*, 7, 713–1018. doi:10.1016/S1570-8659(00)07005-8
- Fei, Y., Rong, G., Wang, B. & Wang, W. (2014). Parallel L-BFGS-B algorithm on GPU. *Computers and Graphics*, 40(1), 1–9. doi:10.1016/j.cag.2014.01.002
- Ferreau, H. J., Kirches, C., Potschka, A., Bock, H. G. & Diehl, M. (2014). qpOASES: a parametric active-set algorithm for quadratic programming. *Mathematical Programming Computation*, 6(4), 327–363. doi:10.1007/s12532-014-0071-1
- Fietzke, B., Mihalyovics, J., King, R. & Peitsch, D. (2021). Binary Repetitive Model Predictive Active Flow Control Applied to an Annular Compressor Stator Cascade with Periodic Disturbances. *Journal of Engineering for Gas Turbines and Power*, (GT2021-58744). doi:10.1115/1.4052382
- Forsgren, A., Gill, P. E. & Wong, E. (2016). Primal and dual active-set methods for convex quadratic programming. *Mathematical Programming*, 159(1-2), 469–508. doi:10.1007/s10107-015-0966-2
- Giuliani, A. & Krivodonova, L. (2019). On the optimal CFL number of SSP methods for hyperbolic problems. *Applied Numerical Mathematics*, 135, 165–172. doi:10.1016/j.apnum.2018.08.015

- Glorot, X. & Bengio, Y. (2010). Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics* (S. 249–256). Zugriff unter <http://proceedings.mlr.press/v9/glorot10a/glorot10a.pdf>
- Gluck, T., Blank, M., Buchl, D. & Kugi, A. (2018). Convex Constrained Iterative Learning Control Using Projection: Application to a Smart Power Switch. *IEEE Transactions on Control Systems Technology*, 26(5), 1818–1825. doi:10.1109/TCST.2017.2734041
- Gorguis, A. (2006). A comparison between Cole-Hopf transformation and the decomposition method for solving Burgers’ equations. *Applied Mathematics and Computation*, 173(1), 126–136. doi:10.1016/j.amc.2005.02.045
- Gray, J. A. T., Vinkeloe, J., Moeck, J., Paschereit, C. O., Stathopoulos, P., Berndt, P. & Klein, R. (2016). Thermodynamic Evaluation of Pulse Detonation Combustion for Gas Turbine Power Cycles. In *Proceedings of ASME Turbo Expo 2016: Turbomachinery technical conference and exposition* (GT2016-57813, V04BT04A044). doi:10.1115/GT2016-57813
- Gugercin, S. & Antoulas, A. C. (2004). A Survey of Model Reduction by Balanced Truncation and Some New Results. *International Journal of Control*, 77(8), 748–766. doi:10.1080/00207170410001713448
- Gunzburger, M. D. & Manservigi, S. (2000). The Velocity Tracking Problem for Navier-Stokes Flows With Boundary Control. *SIAM Journal on Control and Optimization*, 39(2), 1481–1512. doi:10.1137/S0363012999353771
- Gurobi Optimization. (2020). *Gurobi optimizer reference manual*. Version 9.0. Zugriff unter <https://www.gurobi.com/documentation/9.0/refman/index.html>
- Habicht, F., Yücel, F. C., Hanraths, N., Djordjevic, N. & Paschereit, C. O. (2021). Lean Operation of a Pulse Detonation Combustor by Fuel Stratification. *Journal of Engineering for Gas Turbines and Power*, 143(5), 051009. doi:10.1115/1.4048775
- Haghdoust, M. R., Edgington-Mitchell, D. M., Paschereit, C. O. & Oberleithner, K. (2019). Investigation of the Exhaust Flow of a Pulse Detonation Combustor at different Operating Conditions based on High-Speed Schlieren and PIV. *AIAA Scitech 2019 Forum*, 1–15. doi:10.2514/6.2019-1512
- Hairer, E. & Wanner, G. (1999). Stiff differential equations solved by Radau methods. *Journal of Computational and Applied Mathematics*, 111(1-2), 93–111. doi:10.1016/s0377-0427(99)00134-x
- Harten, A., Lax, P. D. & Van Leer, B. (1983). On Upstream Differencing and Godunov-Type Schemes for Hyperbolic Conservation Laws. *SIAM Review*, 25(1), 35–61. doi:10.1137/1025002
- Hasegawa, K., Fukami, K., Murata, T. & Fukagata, K. (2020). CNN-LSTM based reduced order modeling of two-dimensional unsteady flows around a circular cylinder at different Reynolds numbers. *Fluid Dynamics Research*, 52(6), 065501. doi:10.1088/1873-7005/abb91d
- Horlock, J. H. (2013). *Advanced gas turbine cycles: a brief review of power generation thermodynamics*. Elsevier.
- Horlock, J. H. & Marsh, H. (1971). Flow Models for Turbomachines. *Journal of Mechanical Engineering Science*, 13(5), 358–368. doi:10.1243/JMES_JOUR_1971_013_054_02
- Iollo, A., Lanteri, S. & Désidéri, J. A. (2000). Stability Properties of POD-Galerkin Approximations for the Compressible Navier-Stokes Equations. *Theoretical and Computational Fluid Dynamics*, 13, 377–396. doi:10.1007/s001620050119
- Jansohn, P. (2013). *Modern gas turbine systems: High efficiency, low emission, fuel flexible power generation*. Elsevier.

- Julier, S. J. & Uhlmann, J. K. (2004). Unscented filtering and nonlinear estimation. *Proceedings of the IEEE*, 92(3), 401–422. doi:10.1109/jproc.2003.823141
- Kailasanath, K. (2003). Recent Developments in the Research on Pulse Detonation Engines. *AIAA Journal*, 41(2), 145–159. doi:10.2514/2.1933
- Kalman, R. E. (1960). A New Approach to Linear Filtering and Prediction Problems. *Journal of Basic Engineering*, 82(1), 35–45. doi:10.1115/1.3662552
- Karevan, Z. & Suykens, J. A. K. (2020). Transductive LSTM for time-series prediction: An application to weather forecasting. *Neural Networks*, 125, 1–9. doi:10.1016/j.neunet.2019.12.030
- Kasahara, J., Hirano, M., Matsuo, A., Daimon, Y. & Endo, T. (2009). Thrust Measurement of a Multicycle Partially Filled Pulse Detonation Rocket Engine. *Journal of Propulsion and Power*, 25(6), 1281–1290. doi:10.2514/1.42224
- Khong, S. Z., Nešić, D. & Krstić, M. (2016). Iterative learning control based on extremum seeking. *Automatica*, 66, 238–245. doi:10.1016/j.automatica.2015.12.019
- Kiesner, M. & King, R. (2015). Closed-Loop Active Flow Control of the Wake of a Compressor Blade by Trailing-Edge Blowing. In *Proceedings of ASME Turbo Expo 2015: Turbomachinery technical conference and exposition* (GT2015-42026). doi:10.1115/GT2015-42026
- Kiesner, M. & King, R. (2017). Multivariable Closed-Loop Active Flow Control of a Compressor Stator Cascade. *AIAA Journal*, 55(10), 3371–3380. doi:10.2514/1.J055728
- Kim, H. Y. & Won, C. H. (2018). Forecasting the volatility of stock price index: A hybrid model integrating LSTM with multiple GARCH-type models. *Expert Systems with Applications*, 103(1), 25–37. doi:10.1016/j.eswa.2018.03.002
- Kimura, R. (2002). Numerical weather prediction. *Journal of Wind Engineering and Industrial Aerodynamics*, 90(12–15), 1403–1414. doi:10.1016/s0167-6105(02)00261-1
- Krentel, D., Muminovic, R., Brunn, A., Nitsche, W. & King, R. (2010). Application of Active Flow Control on Generic 3D Car Models. In R. King (Hrsg.), *Active Flow Control II. Notes on Numerical Fluid Mechanics and Multidisciplinary Design* (Bd. 108, S. 223–239). doi:10.1007/978-3-642-11735-0_15
- Lange, A. & King, R. (2020). Discrete-Step, Quasi-Newton Extremum Seeking Control for Multivariable Real-Time Optimization. *IFAC-PapersOnLine*, 53(2), 1608–1613. doi:10.1016/j.ifacol.2020.12.2206
- Larochelle, H., Bengio, Y., Louradour, J. & Lamblin, P. (2009). Exploring Strategies for Training Deep Neural Networks. *Journal of Machine Learning Research*, 10(1), 1–40. Zugriff unter <https://www.jmlr.org/papers/volume10/larochelle09a/larochelle09a.pdf>
- Lee, J. H. (2011). Model predictive control: Review of the three decades of development. *International Journal of Control, Automation and Systems*, 9(3), 415–424. doi:10.1007/s12555-011-0300-6
- Lemarie Rieusset, P. G. (2016). *The Navier-Stokes problem in the 21st century* (1. Aufl.). Chapman and Hall/CRC. doi:10.1201/b19556
- LeVeque, R. J. (2002). *Finite Volume Methods for Hyperbolic Problems*. Cambridge University Press. doi:10.1017/CBO9780511791253
- Liang, Y. C., Lee, H. P., Lim, S. P., Lin, W. Z., Lee, K. H. & Wu, C. G. (2002). Proper orthogonal decomposition and its applications - Part I: Theory. *Journal of Sound and Vibration*, 252(3), 527–544. doi:10.1006/jsvi.2001.4041
- Liao, C.-K., Manzie, C., Chapman, A. & Alpcan, T. (2019). Constrained extremum seeking of a MIMO dynamic system. *Automatica*, 108, 108496. doi:10.1016/j.automatica.2019.108496

- Liao, T. W. (2005). Clustering of time series data - A survey. *Pattern Recognition*, 38(11), 1857–1874. doi:10.1016/j.patcog.2005.01.025
- Lindemann, B., Müller, T., Vietz, H., Jazdi, N. & Weyrich, M. (2021). A survey on long short-term memory networks for time series prediction. *Procedia CIRP*, 99, 650–655. doi:10.1016/j.procir.2021.03.088
- Litke, P., Schauer, F., Paxson, D., Bradley, R. & Hoke, J. (2005). Assessment of the Performance of a Pulsejet and Comparison with a Pulsed-Detonation Engine. In *43rd AIAA Aerospace Sciences Meeting and Exhibit* (AIAA 2005-228). doi:10.2514/6.2005-228
- Liu, B., Wang, L., Luo, Z. B., Xia, Z. & Deng, X. (2012). Review of actuators for high speed active flow control. *Science China Technological Sciences*, 55, 2225–2240. doi:10.1007/s11431-012-4861-2
- Lomakin, A. & Korsunov, N. (2019). The Universal Neural Network Model for Solving the Navier-Stokes Equation. In *10th IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications (IDAACS)* (19213282, S. 245–250). doi:10.1109/IDAACS.2019.8924397
- Marchand, H., Martin, A., Weismantel, R. & Wolsey, L. (2002). Cutting planes in integer and mixed integer programming. *Discrete Applied Mathematics*, 123(1-3), 397–446. doi:10.1016/S0166-218X(01)00348-1
- Martin, A. (2001). General Mixed-Integer Programming: Computational Issues for Branch-and-Cut Algorithms. In M. Jünger & D. Naddef (Hrsg.), *Computational Combinatorial Optimization* (Bd. 2241). doi:10.1007/3-540-45586-8_1
- Mathworks. (2021). Matlab 'fmincon' Optimierer. Zugriff unter <https://de.mathworks.com/help/optim/ug/fmincon.html>
- Mayne, D. Q., Rawlings, J. B., Rao, C. V. & Scokaert, P. O. M. (2000). Constrained model predictive control: Stability and optimality. *Automatica*, 36(6), 789–814. doi:10.1016/S0005-1098(99)00214-9
- McElhoe, B. A. (1966). An Assessment of the Navigation and Course Corrections for a Manned Flyby of Mars or Venus. *IEEE Transactions on Aerospace and Electronic Systems*, AES-2(4), 613–623. doi:10.1109/TAES.1966.4501892
- McNally, J., Fernandez, E., Robertson, G., Kumar, R., Taira, K., Alvi, F., ... Murayama, K. (2015). Drag reduction on a flat-back ground vehicle with active flow control. *Journal of Wind Engineering and Industrial Aerodynamics*, 145, 292–303. doi:10.1016/j.jweia.2015.03.006
- Melton, L. P., Koklu, M., Andino, M. & Lin, J. C. (2018). Active Flow Control via Discrete Sweeping and Steady Jets on a Simple-Hinged Flap. *AIAA Journal*, 56(8), 2961–2973. doi:10.2514/1.J056841
- Mills, G. & Krstic, M. (2014). Constrained extremum seeking in 1 dimension. In *Proceedings of the 53rd IEEE Conference on Decision and Control* (S. 2654–2659). doi:10.1109/CDC.2014.7039795
- Mishra, S., Topcu, U. & Tomizuka, M. (2011). Optimization-Based Constrained Iterative Learning Control. *IEEE Transactions on Control Systems Technology*, 19(6), 1613–1621. doi:10.1109/TCST.2010.2083663
- Morari, M., Garcia, C. E. & Prett, D. M. (1988). Model predictive control: Theory and practice. *IFAC Proceedings Volumes*, 21(4), 1–12. doi:10.1016/b978-0-08-035735-5.50006-1
- Morrison, D. R., Jacobson, S. H., Sauppe, J. J. & Sewell, E. C. (2016). Branch-and-bound algorithms: A survey of recent advances in searching, branching, and pruning. *Discrete Optimization*, 19, 79–102. doi:10.1016/j.disopt.2016.01.005

- Moukalled, F., Mangani, L. & Darwish, M. (2015). *The Finite Volume Method in Computational Fluid Dynamics*. Springer Berlin.
- Neddermeyer, F. & King, R. (2019). Advanced nonlinear multi-layer process control for autotrophic cultivations. *IFAC-PapersOnLine*, 52(1), 733–738. doi:10.1016/j.ifacol.2019.06.150
- Nicholls, J. A., Wilkinson, H. R. & Morisson, R. B. (1957). Intermittent Detonation as a Thrust-Producing Mechanism. *Journal of Jet Propulsion*, 27(5), 534–541. doi:10.2514/8.12851
- Nocedal, J. & Wright, S. (2006). *Numerical optimization*. Springer Science & Business Media.
- Oertel, H., Böhle, M. & Reviol, T. (2015). *Strömungsmechanik für Ingenieure und Naturwissenschaftler*. Springer Fachmedien Wiesbaden.
- Owens, D. H. & Daley, S. (2008). Iterative Learning Control - Monotonicity and Optimization. *International Journal of Applied Mathematics and Computer Science*, 18(3), 279–293. doi:10.2478/v10006-008-0026-7
- Parlitz, U. (1998). Nonlinear Time-Series Analysis. In J. A. K. Suykens & J. Vandewalle (Hrsg.), *Nonlinear Modeling* (S. 209–239). Springer. doi:10.1007/978-1-4615-5703-6_8
- Pastoor, M., Henning, L., Noack, B. R., King, R. & Tadmor, G. (2008). Feedback shear layer control for bluff body drag reduction. *Journal of fluid mechanics*, 608, 161–196.
- Peng, C., Sun, L., Zhang, W. & Tomizuka, M. (2016). Optimization-based constrained iterative learning control with application to building temperature control systems. In *Proceedings of IEEE/ASME International Conference on Advanced Intelligent Mechatronics, AIM* (S. 709–715). doi:10.1109/AIM.2016.7576851
- Pólik, I. & Terlaky, T. (2010). Interior Point Methods for Nonlinear Optimization. In G. Di Pillo & F. Schoen (Hrsg.), *Nonlinear Optimization* (Bd. 1989, S. 215–276). Springer. doi:10.1007/978-3-642-11339-0_4
- Poullikkas, A. (2005). An overview of current and future sustainable gas turbine technologies. *Renewable and Sustainable Energy Reviews*, 9(5), 409–443. doi:10.1016/j.rser.2004.05.009
- Pulliam, T. H. (1986). Artificial dissipation models for the Euler equations. *ALAA Journal*, 24(12), 1931–1940. doi:10.2514/6.1985-438
- Qin, S. J. & Badgwell, T. A. (2003). A survey of industrial model predictive control technology. *Control Engineering Practice*, 11(7), 733–764. doi:10.1016/S0967-0661(02)00186-7
- Raissi, M. (2018). Deep hidden physics models: Deep learning of nonlinear partial differential equations. *Journal of Machine Learning Research*, 19, 1–24. Zugriff unter <https://arxiv.org/abs/1801.06637>
- Raissi, M., Perdikaris, P. & Karniadakis, G. E. (2019). Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *Journal of Computational Physics*, 378, 686–707. doi:10.1016/j.jcp.2018.10.045
- Raissi, M., Wang, Z., Triantafyllou, M. S. & Karniadakis, G. E. (2019). Deep learning of vortex-induced vibrations. *Journal of Fluid Mechanics*, 861, 119–137. doi:10.1017/jfm.2018.872
- Ramm, A. G. (2017). Global existence, uniqueness and estimates of the solution to the Navier–Stokes equations. *Applied Mathematics Letters*, 74, 154–160. doi:10.1016/j.aml.2017.05.009
- Rashad, R., Califano, F., Schuller, F. P. & Stramigioli, S. (2021). Port-Hamiltonian modeling of ideal fluid flow: Part I. Foundations and kinetic energy. *Journal of Geometry and Physics*, 164, 104201. doi:10.1016/j.geomphys.2021.104201

- Rawlings, J., Meadows, E. & Muske, K. (1994). Nonlinear Model Predictive Control: A Tutorial and Survey. *IFAC Proceedings Volumes*, 27(2), 185–197. doi:10.1016/s1474-6670(17)48151-1
- Reichel, T. G., Schäpel, J.-S., Bobusch, B. C., Klein, R., King, R. & Paschereit, C. O. (2016). Shockless Explosion Combustion: Experimental Investigation of a New Approximate Constant Volume Combustion Process. *Journal of Engineering for Gas Turbines and Power*, 139(2), 021504. doi:10.1115/1.4034214
- Reiss, J., Schulze, P., Sesterhenn, J. & Mehrmann, V. (2018). The Shifted Proper Orthogonal Decomposition: A Mode Decomposition for Multiple Transport Phenomena. *SIAM Journal on Scientific Computing*, 40(3), A1322–A1344. doi:10.1137/17M1140571
- Robinson, J. C., Rodrigo, J. L. & Sadowski, W. (2016). *The Three-Dimensional Navier–Stokes Equations: Classical Theory*. Cambridge University Press.
- Roe, P. L. (1986). Characteristic-Based Schemes for the Euler Equations. *Annual Review of Fluid Mechanics*, 1, 337–365. doi:10.1146/annurev.fluid.18.1.337
- Rouser, K., King, P., Schauer, F., Sondergaard, R. & Hoke, J. (2010). Unsteady Performance of a Turbine Driven by a Pulse Detonation Engine. In *48th AIAA Aerospace Sciences Meeting Including the New Horizons Forum and Aerospace Exposition* (2010-1116). doi:10.2514/6.2010-1116
- Rowley, C. W., Colonius, T. & Murray, R. M. (2004). Model reduction for compressible flows using POD and Galerkin projection. *Physica D: Nonlinear Phenomena*, 189(1-2), 115–129. doi:10.1016/j.physd.2003.03.001
- Rowley, C. W. & Dawson, S. T. (2017). Model Reduction for Flow Analysis and Control. *Annual Review of Fluid Mechanics*, 49(August), 387–417. doi:10.1146/annurev-fluid-010816-060042
- Sager, S., Bock, H. G. & Diehl, M. (2007). Solving Mixed–Integer Control Problems by Sum Up Rounding With Guaranteed Integer Gap. *Universität Heidelberg*. doi:10.11588/heidok.00008384
- Schäpel, J.-S., King, R., Yücel, F., Völzke, F., Paschereit, C. O. & Klein, R. (2018). Fuel Injection Control for a Valve Array in a Shockless Explosion Combustor. In *ASME Turbo Expo 2018: Turbomachinery Technical Conference and Exposition* (GT2018-75295, V006T05A007). doi:10.1115/GT2018-75295
- Schäpel, J.-S., Wolff, S., Schulze, P., Berndt, P., Klein, R., Mehrmann, V. & King, R. (2017). State estimation for reactive Euler equation by Kalman Filtering. *CEAS Aeronautical Journal*, 8(2), 261–270. doi:10.1007/s13272-016-0229-9
- Schneider, R. & Georgakis, C. (2013). How to NOT Make the Extended Kalman Filter Fail. *Industrial and Engineering Chemistry Research*, 52(9), 3354–3362. doi:10.1021/ie300415d
- Serre, G., Lafon, P., Gloerfelt, X. & Bailly, C. (2012). Reliable reduced-order models for time-dependent linearized Euler equations. *Journal of Computational Physics*, 231(15), 5176–5194. doi:10.1016/j.jcp.2012.04.019
- Sherstinsky, A. (2020). Fundamentals of Recurrent Neural Network (RNN) and Long Short-Term Memory (LSTM) network. *Physica D: Nonlinear Phenomena*, 404, 132306. doi:10.1016/j.physd.2019.132306
- Shi, D., Zhou, J. & Shi, D. (2013). A new low order least squares nonconforming characteristics mixed finite element method for Burgers’ equation. *Applied Mathematics and Computation*, 219(24), 11302–11310. doi:10.1016/j.amc.2013.05.037
- Shinde, V., Longatte, E., Baj, F., Hoarau, Y. & Braza, M. (2016). A Galerkin-free model reduction approach for the Navier-Stokes equations. *Journal of Computational Physics*, 309, 148–163. doi:10.1016/j.jcp.2015.12.051

- Smith, G. L., Schmidt, S. F. & McGee, L. A. (1962). *Application of Statistical Filter Theory to the Optimal Estimation of Position and Velocity on Board a Circumlunar Vehicle*. National Aeronautics and Space Administration. Zugriff unter <https://ntrs.nasa.gov/citations/20190002215>
- Spurk, J., Schobeiri, T. & Marschall, H. (2012). *Fluid Mechanics: Problems and Solutions*. Springer Berlin Heidelberg.
- Stathopoulos, P. (2018). Comprehensive Thermodynamic Analysis of the Humphrey Cycle for Gas Turbines with Pressure Gain Combustion. *Energies*, 11(12), 3521. doi:10.3390/en11123521
- Stathopoulos, P. (2020). An alternative architecture of the Humphrey cycle and the effect of fuel type on its efficiency. *Energy Science and Engineering*, 8(10), 3702–3716. doi:10.1002/ese3.776
- Steinberg, S. J., King, R., Staats, M. & Nitsche, W. (2015). Iterative Learning Active Flow Control Applied to a Compressor Stator Cascade With Periodic Disturbances. *Journal of Turbomachinery*, 137(11), 111003. doi:10.1115/1.4031251
- Steinberg, S. J., Staats, M., Nitsche, W. & King, R. (2017). Comparison of Conventional and Repetitive MPC with Application to a Periodically Disturbed Compressor Stator Vane Flow. *IFAC-PapersOnLine*, 50(1), 11107–11112. doi:10.1016/j.ifacol.2017.08.948
- Takapoui, R., Moehle, N., Boyd, S. & Bemporad, A. (2020). A simple effective heuristic for embedded mixed-integer quadratic programming. *International Journal of Control*, 93(1), 2–12. doi:10.1080/00207179.2017.1316016
- Temam, R. (2001). *Navier-Stokes equations: theory and numerical analysis*. American Mathematical Soc.
- Topalovic, D., Arnold, F. & King, R. (2022). Reduction of Pressure Fluctuations in an Annular Pulsed Detonation Combustor Mockup by Iterative Learning Control Using Eigenvector-Based Binary Solution Sets and Iterative Model Identification. In R. King & D. Peitsch (Hrsg.), *Active Flow and Combustion Control 2018. Notes on Numerical Fluid Mechanics and Multidisciplinary Design* (Bd. 152, S. 116–131). Springer. doi:10.1007/978-3-030-90727-3_8
- Toro, E. F. (2013). *Riemann solvers and numerical methods for fluid dynamics: a practical introduction*. Springer Science & Business Media.
- van der Schaft, A. (2006). Port-Hamiltonian Systems: An Introductory Survey. In *Proceedings of the international congress of mathematicians* (Bd. 3, S. 1339–1365). Zugriff unter <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.366.3380&rep=rep1&type=pdf>
- van der Schaft, A. & Jeltsema, D. (2014). Port-Hamiltonian systems theory: An introductory overview. *Foundations and Trends in Systems and Control*, 1(2-3), 173–378. doi:10.1561/26000000002
- Vazquez, R., Schuster, E. & Krstic, M. (2007). A closed-form feedback controller for stabilization of magnetohydrodynamic channel flow. In *2007 European Control Conference, ECC 2007* (S. 5572–5579). doi:10.23919/ECC.2007.7068273
- Vazquez, S., Leon, J. I., Franquelo, L. G., Rodriguez, J., Young, H. A., Marquez, A. & Zanchetta, P. (2014). Model Predictive Control: A Review of its Applications in Power Electronics. *IEEE Industrial Electronics Magazine*, 8(1), 16–31. doi:10.1109/MIE.2013.2290138
- Vielma, J. P., Dunning, I., Huchette, J. & Lubin, M. (2017). Extended formulations in mixed integer conic quadratic programming. *Mathematical Programming Computation*, 9(3), 369–418. doi:10.1007/s12532-016-0113-y
- Walsh, P. P. & Fletcher, P. (2004). *Gas turbine performance*. John Wiley & Sons.

- Whalen, E. A., Shmilovich, A., Spoor, M., Tran, J., Vijgen, P., Lin, J. C. & Andino, M. (2018). Flight Test of an Active Flow Control Enhanced Vertical Tail. *AIAA Journal*, 56(9), 3393–3398. doi:10.2514/1.J056959
- Willcox, K. & Perairet, J. (2002). Balanced model reduction via the proper orthogonal decomposition. *AIAA Journal*, 40(11), 2323–2330. doi:10.2514/6.2001-2611
- Wolff, S. (2018). *Entwicklung von Fehlererkennungs-, Zustandsschätzungs-, Regelungs- und ganzzahligen Optimalsteuerungsmethoden für pulsierende, detonative Brennkammern anhand eines akustischen Ersatzsystems* (Diss., Technische Universität Berlin).
- Wolff, S. & King, R. (2015). An Annular Pulsed Detonation Combustor Mockup: System Identification and Misfiring Detection. *Journal of Engineering for Gas Turbines and Power*, 138(4), 041603. doi:10.1115/1.4031320
- Wolff, S., Schäpel, J.-S. & King, R. (2016). Application of Artificial Neural Networks for Misfiring Detection in an Annular Pulsed Detonation Combustor Mockup. *Journal of Engineering for Gas Turbines and Power*, 139(4), 041510. doi:10.1115/1.4034941
- Yücel, F. C., Habicht, F., Arnold, F., King, R., Bohon, M. & Paschereit, C. O. (2021). Controlled autoignition in stratified mixtures. *Combustion and Flame*, 232, 111533. doi:10.1016/j.combustflame.2021.111533
- Yücel, F. C., Habicht, F., Jaeschke, A., Lückoff, F., Oberleithner, K. & Paschereit, C. O. (2021). Investigation of the Fuel Distribution in a Shockless Explosion Combustor. *Journal of Engineering for Gas Turbines and Power*, 143(1), 011008. doi:10.1115/1.4049220
- Yücel, F. C., Völzke, F. & Paschereit, C. O. (2019). Effect of the Switching Times on the Operating Behavior of a Shockless Explosion Combustor. In R. King (Hrsg.), *Active Flow and Combustion Control 2018. Notes on Numerical Fluid Mechanics and Multidisciplinary Design* (Bd. 141, S. 121–134). Springer. doi:10.1007/978-3-319-98177-2_8
- Zel'dovich, Y. B. (1940). On the use of detonative combustion in power engineering. *Journal of Technical Physics*, 10(17), 1453–1461.
- Zhang, L. & Suganthan, P. N. (2016). A survey of randomized algorithms for training neural networks. *Information Sciences*, 364–365, 146–155. doi:10.1016/j.ins.2016.01.039