

**Band  
430**

Verlagsschriftenreihe des Heinz Nixdorf Instituts  
Dr.-Ing. Julia Timmermann (Hrsg.)  
Regelungstechnik und Mechatronik

Julia Timmermann, Ricarda-Samantha Götte,  
Annika Junker, Michael Hesse, Luis Schwarzer

# **DART – Datengetriebene Methoden in der Regelungstechnik**



***Julia Timmermann***

***Ricarda-Samantha Götte***

***Michael Hesse***

***Annika Junker***

***Luis Schwarzer***

***DART – Datengetriebene Methoden in  
der Regelungstechnik***

***DART – Data-driven Methods in Con-  
trol***

**Bibliografische Information Der Deutschen Bibliothek**

Die Deutsche Bibliothek verzeichnet diese Publikation in der Deutschen Nationalbibliografie; detaillierte bibliografische Daten sind im Internet über <http://dnb.ddb.de> abrufbar.

Band 430 der Verlagsschriftenreihe des Heinz Nixdorf Instituts

© Heinz Nixdorf Institut, Universität Paderborn – Paderborn – 2025

ISSN (Online): 2365-4422

ISBN: 978-3-947647-49-1

Das Werk einschließlich seiner Teile ist urheberrechtlich geschützt. Jede Verwertung außerhalb der engen Grenzen des Urheberrechtsgesetzes ist ohne Zustimmung der Herausgeber und des Verfassers unzulässig und strafbar. Das gilt insbesondere für Vervielfältigung, Übersetzungen, Mikroverfilmungen, sowie die Einspeicherung und Verarbeitung in elektronischen Systemen.

Als elektronische Version frei verfügbar über die Digitalen Sammlungen der Universitätsbibliothek Paderborn.

Satz und Gestaltung: Julia Timmermann



*Data without context is just noise.*



# Danksagung

Wissenschaftliche Forschungsprojekte können nur dann in einem größeren Umfang durchgeführt werden, wenn ein Fördergeber die Ideen des Projektes positiv bewertet und damit auch finanziell fördert. Im Rahmen der Förderung von KI-Nachwuchswissenschaftlerinnen durch das Bundesministerium für Bildung und Familie (BMBF) war es mir als Projektleiterin möglich zusammen mit meinem Team über vier Jahre zu relevanten Themen im Bereich der datengetriebenen Methoden in der Regelungstechnik zu forschen. Für diese Möglichkeit und auch die damit verbundenen Freiheiten in der Grundlagenforschung bedanken wir uns ganz herzlich. Die fachliche Betreuung wurde vom Projektträger Deutsches Zentrum für Luft- und Raumfahrt (DLR) in Berlin übernommen. Unser Dank geht daher an Frau Dr. Wunram vom DLR, die uns in dieser Zeit immer gut beraten hat und unsere Forschungsarbeit unterstützt hat. Auch für die organisatorische Hilfe und Unterstützung bedanken wir uns mit den weiteren Mitarbeiter\*innen des DLR, mit denen wir in Kontakt waren.

Als Leiterin der Nachwuchsforschungsgruppe DART geht mein größter Dank an mein Team an wissenschaftlichen Mitarbeiter\*innen: Dr.-Ing. Ricarda-Samantha Götte, Dr.-Ing. Annika Junker und Dr.-Ing. Michael Hesse waren das feste Team, das während der Projektlaufzeit an Inhalten, Veröffentlichungen, Meilensteinen und allem weiteren im Kontext von DART gearbeitet hat. Außerdem bedanke ich mich bei Luis Schwarzer, der das Team im Rahmen des Projektabschlusses unterstützt hat und vor allem die Arbeiten am Demonstrator verantwortet hat. Als Projektleiterin hatte ich zum einen die Verantwortung der Organisation der Arbeiten im Projekt und konnte dadurch Einfluss auf Inhalte und den Fortschritt der Forschung nehmen. Zum anderen war es immer eine tolle Zusammenarbeit auch die Forschungsarbeiten der einzelnen Mitarbeitenden gut zu betreuen und die selbstständige wissenschaftliche Arbeit bis hin zu Promotion aller drei Doktorand\*innen zu fördern. Die guten Projektergebnisse konnten nur durch die tolle Teamarbeit entstehen und ich bedanke mich bei allen ganz herzlich dafür.

Die Nachwuchsgruppe ist natürlich nicht unabhängig vom weiteren Institutionen an der Universität Paderborn zu sehen. Sie ist aus dem Lehrstuhl für Regelungstechnik und Mechatronik (RtM) geleitet von Prof. Ansgar Trächtler hervorgegangen. Daher bedanken wir uns ganz herzlich bei Prof. Trächtler, dass wir von ihm gefördert und unterstützt wurden. Zunächst konnten wir im Rahmen unserer Tätigkeit am RtM die Verwendung von Methoden des maschinellen Lernens innerhalb der Regelungstechnik kennenlernen und den Antrag von DART schreiben. Aber auch während der Projektlaufzeit hat Prof. Trächtler uns mit seiner Expertise und der Teilnahme an Projekt- und Meilensteintreffen unterstützt. Die Diskussionen über Forschungsideen war dabei immer konstruktiv und fruchtbar. Es haben uns aber auch weitere Personen des RtM geholfen. Im Labor waren Martin Leibenger und Jörg Schaffrath immer hilfreiche Ansprechpartner und konnten mit Wissen und

Werkzeug weiterhelfen. Bei einem so großen Forschungsprojekt mit mehreren Mitarbeitenden und Studierenden fallen auch viele organisatorische und Verwaltungsaufgaben an. Bei Michaela Wiemers aus dem Sekretariat vom RtM möchten wir uns daher bedanken, dass sie immer geduldig unsere Fragen zu Dienstreiseabrechnungen und anderer Uni-Bürokratie beantwortet hat. Außerdem hat Heike Rieger die Finanzverwaltung von DART übernommen und war dabei immer eine zuverlässige Ansprechpartnerin für alle finanziellen Aufgaben. Vielen Dank an alle genannten Personen für die großartige Unterstützung in den vier Jahren der Forschungsarbeiten im Projekt DART.

Im Rahmen des DART-Projektes haben wir auch an vielen Stellen mit Studierenden der Universität Paderborn hauptsächlich aus dem Studiengang Maschinenbau zusammengearbeitet. Insgesamt haben uns zehn Studenten als studentische Hilfskräfte (mit Bachelor) bei den Forschungsarbeiten unterstützt. Dabei ging es um die Implementierung von Algorithmen aber hauptsächlich auch praktische Arbeiten im Labor. Dies beinhaltete Experimente an Prüfständen bzw. auch die Mitarbeit bei der Entwicklung und dem Aufbau unseres eigenen Demonstrators. Außerdem haben wir 24 studentische Arbeiten mit Themen aus dem DART Kontext betreuen können, in denen die Studierenden unterschiedlichste Aufgaben bearbeitet haben. Allen Studierenden danken wir für ihre motivierte Arbeit und ihr Interesse an unseren Forschungsthemen und hoffen, dass sie ihre Erkenntnisse aus dem Projekt weitertragen können.

Im Namen des gesamten DART-Teams  
Dr.-Ing. Julia Timmermann

# Vorwort

Das Projekt *DART - Datengetriebene Methoden in der Regelungstechnik* entstand im Rahmen der Förderung von KI-Nach Nachwuchswissenschaftlerinnen, wozu es im Jahr 2019 einen Aufruf vom Bundesministerium für Bildung und Forschung (BMBF) gab. Das Ziel dieser Förderlinie war es, die Beteiligung von Frauen in der deutschen Forschung zur künstlichen Intelligenz zu erhöhen und ihnen außerdem akademische Führungspositionen zu ermöglichen. Es wurde festgestellt, dass die KI-Forschung nicht nur kompetente Fachleute in Fachgebieten wie Informatik, Mathematik oder Kognitionspsychologie, sondern zunehmend auch Fachkompetenzen aus den verschiedensten Anwendungsgebieten benötigt. Um dies zu erreichen, wurden Frauen, die im Bereich von KI-Fragestellungen forschen, dazu aufgefordert, sich um eine Nachwuchsgruppe unter der Leitung dieser Frauen, zu bewerben und damit den Anteil von qualifizierten Frauen in Führungspositionen der deutschen KI-Forschung zu steigern. Damit war diese Fördermaßnahme ein Teil der Umsetzung der KI-Strategie der Bundesregierung und der Hightech-Strategie 2025.

DART ist eine Nachwuchsforschungsgruppe, die sich entsprechend der Ausschreibung auf die Anwendung von maschinellem Lernen in einem Anwendungsgebiet, der Regelungstechnik, spezialisiert hat und die Forschungsarbeiten liefen in der Zeit von Juli 2020 bis November 2024 an der Universität Paderborn. Auf Grundlage der klassischen Methoden in der Regelungstechnik ist der Sprung zur Anwendung von maschinellem Lernen sehr naheliegend. Denn in regelungstechnischen Anwendungen werden schon immer Daten z.B. in Form von Messungen verwendet, um das Verhalten des Systems zu beeinflussen oder regeln zu können. Der Trend, den man in sehr vielen Forschungsgemeinschaften sieht, nun vermehrt datenbasierte Lernalgorithmen zu verwenden, kann daher auch in besonderer Form in regelungstechnischen Methoden aufgenommen werden. Beispielsweise arbeiteten Regelungstechniker\*innen auch in der Vergangenheit schon daran, auf Systemveränderungen im Betrieb durch adaptive Methoden zu reagieren. Ähnliche Ziele verfolgen nun auch Lernalgorithmen, die sich aktiv dem Systemverhalten anpassen und dabei Methoden des maschinellen Lernens verwenden.

Daher war es das Ziel der Nachwuchsgruppe *Datengetriebene Methoden in der Regelungstechnik*, die synergetische Kombination modell- und datengetriebener Verfahren für regelungstechnische Aufgaben zu erforschen. Dazu werden modellbasierte Verfahren mit maschinellem Lernen kombiniert, um hybride Methoden zu erhalten und die größtmögliche Performanz beim Regelungsentwurf zu erzielen. Aus der Forschungsgemeinschaft zu maschinellem Lernen ist der Standardansatz bekannt, Systeme oder Methoden zu verwenden, die ausschließlich auf ML basieren. Hier wird immer wieder als Vorteil genannt, dass dadurch kein Vorwissen von Expert\*innen für die Anwendung vorhanden sein muss. Daher wird ML als Alternative zu klassischen Methoden vorgeschlagen. Für die Arbeit der Nachwuchsgruppe nehmen wir aber eine andere Sichtweise an: Das Fachwissen aus dem ingenieurwissenschaftlichen

Kontext ist für uns sehr wertvoll und es gibt keine schlüssigen Gründe, etablierte Methoden komplett durch datenbasierte Ansätze zu ersetzen. Daher streben wir nach einer synergetischen, methodischen Kombination modell- und datengetriebener Verfahren für den Regelungs- und Steuerungsentwurf, um größtmöglichen Nutzen aus beiden Herangehensweisen zu ziehen.

Dieses Buch dient der vollständigen Zusammenfassung der wissenschaftlichen Forschungsergebnisse, die in der vierjährigen Laufzeit der Nachwuchsforschungsgruppe *DART – Datengetriebene Methoden in der Regelungstechnik* erzielt werden konnten. Es beinhaltet die notwendigen Grundlagen aus dem Bereich des maschinellen Lernens und präsentiert auch die Grundlagen der Regelungstechnik, die als Ausgangspunkt für die weiterentwickelten Methoden verwendet werden, ist aber nicht als grundlegendes Lehrbuch für diese Bereiche zu sehen. Es besteht in der Darstellung somit kein Anspruch auf die Vollständigkeit bei den Grundlagen, sondern es ist das Ziel, den Lesenden Handlungsempfehlungen für die Anwendung der entwickelten Methoden zu geben. Somit werden die neuen Methoden ausführlich vorgestellt und ihr jeweiliges Anwendungsgebiet spezifiziert, so dass der Anwendende entscheiden kann, welche Methode für sein Problem geeignet ist.

# Abkürzungsverzeichnis

BO	Bayessche Optimierung
DART	Datengetriebene Methoden in der Regelungstechnik
DMD	Dynamic Mode Decomposition
EDMD	Extended Dynamic Mode Decomposition
EKF	Extended Kalman Filter
GP	Gauß-Prozess
GRU	Gated Recurrent Unit
HNI	Heinz Nixdorf Institut
IDA	Interconnection and Damping Assignment
IMU	Inertia Measurement Unit
InvΓ	Inverse Gamma-Verteilung
JE	Joint Estimation
JE-SRUKF	Joint Estimation Square Root Unscented Kalman Filter
KI	künstliche Intelligenz
LASSO	Least Absolute Shrinkage and Selection Operator
LQR	Linear quadratische Regelung
LSTM	Long Short-Term Memory
ML	maschinelles Lernen
MM	Moment Matching
MOPGRNN	Multi-Objective Physics-Guided Recurrent Neural Network
MPC	modellprädiktive Regelung
NN	neuronales Netz
PCA	Principal Component Analysis
PCH	Port-Controlled Hamilton System
PCHD	Port-Controlled Hamilton System with Dissipation
PGNN	Physics-Guided Neural Network
PGRNN	Physics-Guided Recurrent Neural Network

## *Abkürzungsverzeichnis*

RHS	Regularized-Horseshoe-Verteilung
RL	Reinforcement Learning
RNN	rekurrentes neuronales Netz
RRT*	Rapidly Exploring Random Tree* Algorithmus
SE	Squared Exponential
SINDy	Sparse Identification of Nonlinear Dynamics
SMO	Sliding-Mode-Beobachter
SQP	Sequentielle quadratische Programmierung
SRUKF	Square Root Unscented Kalman Filter
UKF	Unscented Kalman Filter
UT	Unscented Transformation



# Inhaltsverzeichnis

<b>Abkürzungsverzeichnis</b>	<b>ix</b>
<b>1 Einleitung</b>	<b>1</b>
1.1 Motivation und Ziele . . . . .	2
1.2 Forschungsumfeld der Nachwuchsgruppe . . . . .	3
1.3 Motivation für datengetriebene Methoden in der Regelungstechnik . .	6
<b>2 Grundlagen des maschinellen Lernens in der Regelungstechnik</b>	<b>11</b>
2.1 Datengetriebene Methoden . . . . .	11
2.2 Grundlagen des maschinellen Lernens . . . . .	16
2.3 Zielsetzung beim hybriden Regelungsentwurf . . . . .	22
<b>3 Modellbildung</b>	<b>25</b>
3.1 Methoden basierend auf dem Koopman-Operator . . . . .	26
3.1.1 Theoretische Einordnung des Koopman-Operators . . . . .	26
3.1.2 Systemidentifikation mittels des Koopman-Operators . . . . .	29
3.1.3 Dynamic Mode Decomposition . . . . .	29
3.1.4 Extended Dynamic Mode Decomposition . . . . .	32
3.2 Sparse Identification of Nonlinear Dynamical Systems . . . . .	42
3.3 Physikalisch motivierte neuronale Netze . . . . .	45
3.3.1 Neuronale Netze . . . . .	46
3.3.2 Physics-Guided Neural Networks . . . . .	47
3.3.3 Physics-Guided Recurrent Neural Networks . . . . .	52
3.4 Datengetriebene PCHD-Modelle . . . . .	58
3.4.1 PCHD-Modelle . . . . .	58
3.4.2 Algorithmus zur Bestimmung datengetriebener PCHD-Modelle	59
3.4.3 Beispielhafte Anwendung . . . . .	61
3.5 Probabilistische hybride Modellierung mittels Gauß-Prozess-Regression	70
3.5.1 Entwicklung des datengetriebenen Modellteils . . . . .	70
3.5.2 Effiziente probabilistische Zustandspropagation . . . . .	73
<b>4 Steuerungs- und Regelungsentwurf</b>	<b>77</b>
4.1 Riccati-Regelungen mittels des Koopman-Operators . . . . .	77
4.1.1 Illustration der Idee . . . . .	77
4.1.2 Beispielhafte Anwendung . . . . .	81
4.2 Modellprädiktive Regelung mittels des Koopman-Operators . . . . .	82
4.3 Passivitätsbasierte Regelung mittels datengetriebener PCHD-Modelle	86
4.3.1 Beispielhafte Anwendung . . . . .	87
4.4 Entwurfsmethoden im Kontext der Gauß-Prozess-Regression . . . . .	91
4.4.1 Hybride Optimalsteuerung dynamischer Systeme . . . . .	91
4.4.2 Hybride Zustandslinearisierung für eingangs-affine Systeme . .	99

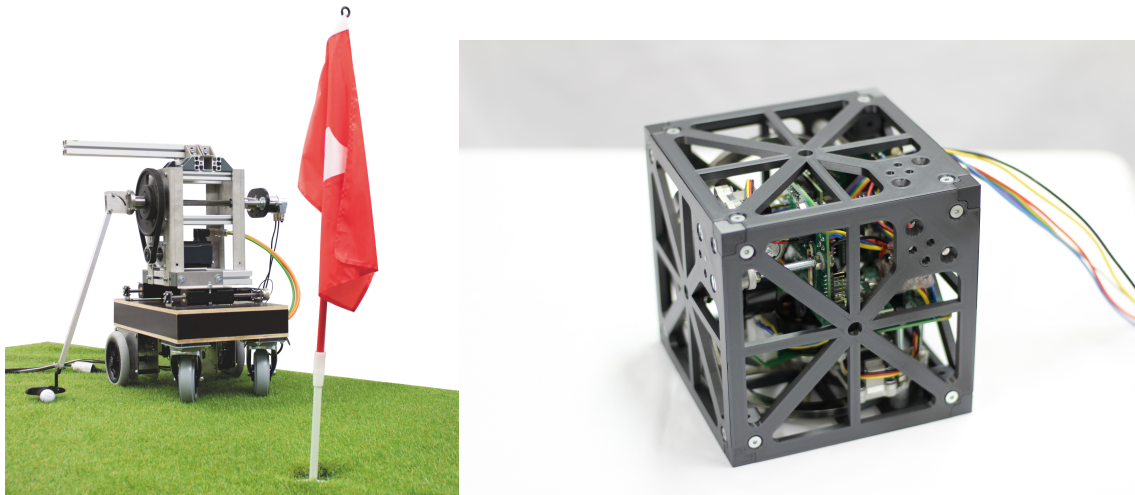
4.4.3	Interaktiver Entwurf mit Bayesscher Optimierung . . . . .	103
<b>5</b>	<b>Beobachterentwurf</b>	<b>109</b>
5.1	Grundlagen zum Beobachterentwurf . . . . .	109
5.2	Joint Estimation . . . . .	111
5.2.1	Augmentierter Beobachterentwurf . . . . .	114
5.2.2	Beispielhafte Anwendungen . . . . .	120
5.2.3	Augmentierter Sliding-Mode-Beobachter . . . . .	129
5.3	Automatische Modellaktualisierung . . . . .	132
5.3.1	Merkmalsextraktion . . . . .	132
5.3.2	Modellaktualisierung . . . . .	135
5.4	Koopman-basierter Beobachter mit Verwendung von EDMD . . . . .	137
<b>6</b>	<b>Demonstratoren</b>	<b>141</b>
6.1	Golfroboter . . . . .	141
6.1.1	Mechatronischer Entwurf des Golfroboters . . . . .	143
6.1.2	Bildverarbeitung . . . . .	152
6.1.3	Bestimmung eines optimalen Schlages . . . . .	153
6.1.4	Ergebnisse . . . . .	156
6.2	Selbstbalancierender Würfel . . . . .	159
6.2.1	Mechatronischer Entwurf des Würfels . . . . .	159
6.2.2	Modellbasierter Regelungsentwurf des Würfels . . . . .	163
	<b>Stichwortverzeichnis</b>	<b>171</b>
	<b>Literaturverzeichnis</b>	<b>173</b>

# 1 Einleitung

Dieses Buch handelt von der Entwicklung von Schnittstellen zwischen datengetriebenen Methoden aus dem Bereich des maschinellen Lernens (ML) und klassischen regelungstechnischen Ansätzen. Dabei werden viele methodische Grundlagen verwendet, die aus unterschiedlichen Bereichen stammen: Optimierung, Stochastik, lineare Algebra, Differentialgleichungen und natürlich die grundlegenden Ideen maschineller Lernmethoden. Dabei hat das Buch nicht den Anspruch, alle Grundlagen zu erklären, gibt aber Hinweise auf die wichtigsten Referenzen aus diesen Bereichen. Da in der klassischen Ingenieurausbildung ML eine geringe bis gar keine Rolle spielt, werden in diesem Bereich mehr grundlegende Methoden erläutert, die dann im komplexeren Setting der hybriden Methoden verwendet werden. Dazu gehören beispielsweise die neuronalen Netze als eine der bekanntesten Strukturen aus dem maschinellen Lernen.

Der Beitrag dieses Buches soll es ermöglichen, Regelungstechnik auch für Systeme anwenden zu können, die im Zuge der fortschreitenden Entwicklungen immer komplexer werden. Dabei ersetzen datengetriebene Methoden nicht die physikalisch mathematische Herleitung und Beschreibung, sondern erweitern diese für eine auch zukünftig erfolgreiche Anwendung. Dies ist möglich, da die Entwicklung des letzten Jahrzehnts zeigt, dass datengetriebene Methoden durch günstiger werdende Sensoren, stärkere Rechenleistung und bessere Speicher- und Transfermöglichkeiten insgesamt zielgerichteter eingesetzt werden können.

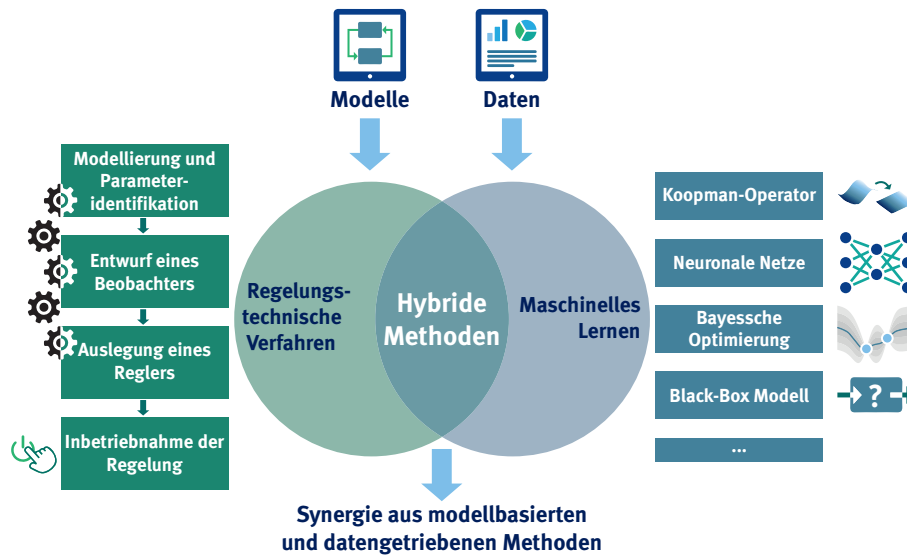
Die Themen dieses Buches orientieren sich am regelungstechnischen Entwurfsprozess. Zunächst werden in Kapitel 2 die Grundlagen des maschinellen Lernens im Kontext der Entwicklung hybrider Methoden in der Regelungstechnik vorgestellt, um in diesem Bereich noch zusätzliches Grundlagenwissen anbieten zu können. Anschließend folgen etablierte und neu entwickelte hybride Methoden aus den Bereichen Modellbildung in Kapitel 3, Steuerungs- und Regelungsentwurf in Kapitel 4 und Beobachterentwurf in Kapitel 5. Abschließend werden die zwei großen Demonstratoren unseres Projektes in Kapitel 6 vorgestellt. Dies sind zum einen der Golfroboter und zum anderen ein selbstbalancierender Würfel, der im Rahmen des Projektes entwickelt und aufgebaut wurde, um die Leistungsfähigkeit der entwickelten Methoden zu untersuchen (siehe Abbildung 1.1). Die Anwendung an realen Demonstratoren war dabei ein sehr wichtiges Ziel, da es einen großen Unterschied bedeutet, Methoden nur simulativ oder experimentell in der realen Welt zu testen. Eine Umsetzung im industriellen Umfeld ist noch nicht umgesetzt worden, aber es wäre ein Wunsch der Nachwuchsgruppe durch den Beitrag dieses Buchs auch hier Interesse zu wecken.



**Abbildung 1.1** Der Golfroboter und der selbstbalancierende Würfel sind zwei wichtige Demonstratoren, an denen die hybriden Methoden der Nachwuchsgruppe experimentell untersucht wurden.

### 1.1 Motivation und Ziele

Im Zuge der Digitalisierung erfahren maschinelles Lernen und datengetriebene Methoden aktuell hohe Aufmerksamkeit seitens Wissenschaft und Industrie. Es fehlen jedoch die Grundlagen und das Verständnis, die datengetriebenen Verfahren der Informatik mit bewährten modellbasierten Verfahren der Ingenieurwissenschaften wie dem modellbasierten Entwurf in der Mechatronik und regelungstechnischen Methoden sinnvoll zu kombinieren, um dadurch hybride Modelle zu erhalten. Diese Ingenieurverfahren beruhen auf physikalischen Verhaltensmodellen, die eine besonders kondensierte und interpretierbare Wissensrepräsentation darstellen und insbesondere kausale Zusammenhänge beschreiben. Dabei gibt es für spezifische regelungstechnische Anwendungen umfangreiches Vorwissen in Form von bekannten Strukturen und Informationen, wie beispielsweise (Teil-) Modelle oder Parametersätze, die auch beim Einsatz von Methoden wie z.B. maschinellem Lernen sinnvollerweise weiterverwendet werden sollten. Eine solche sinnvolle systematische Kombination ist wissenschaftlich noch wenig untersucht, erst recht hinsichtlich einer industriellen Anwendung. Daher war es das Hauptziel der Nachwuchsforschungsgruppe, die synergetische Kombination modell- und datengetriebener Verfahren für regelungstechnische Aufgabenstellungen zu erforschen und dies ist auch in unserer Projektübersicht in Abbildung 1.2 sichtbar. Dabei sollen die Vorteile und Stärken der jeweiligen Verfahren strukturell beibehalten werden: bei den Verfahren mit physikalischen Modellen deren gute Interpretierbarkeit, tiefere Einsicht in das Systemverhalten und die Wiedergabe kausaler Zusammenhänge; bei den datengetriebenen Verfahren deren Fähigkeit, physikalisch schwer beschreibbare Zusammenhänge auch ohne tiefe spezifische Anwendungskenntnisse abbilden zu können und ihre Erweiterbarkeit durch die Möglichkeit auch während des Betriebs weiter zu lernen. Der Schwerpunkt dieses Vorhabens lag somit auf der Entwicklung von hybriden Methoden, die modellgetriebene Verfahren mit maschinellem Lernen kombinieren, um größtmögliche Performanz beim Regelungsentwurf zu erzielen.



**Abbildung 1.2** Projektidee von DART

Das typische Vorgehen beim Regelungsentwurf beinhaltet folgende Teilaufgaben:

1. Modellierung des physikalischen Systemverhaltens mit Dynamikgleichungen,
2. Identifikation der in den Dynamikgleichungen enthaltenen Parameter,
3. Beobachtersauslegung zur Schätzung nicht messbarer, aber für die Regelung benötigter Zustandsgrößen oder Parameter,
4. Entwurf einer Regelung, abhängig von der Art der Dynamikgleichungen zu wählen,
5. Inbetriebnahme des Reglers am Prüfstand.

Bei unserem Ziel der synergetischen, methodischen Kombination modell- und datengetriebener Verfahren für den Regelungs-/ Steuerungsentwurf werden wir diese Schritte der Modellierung und Regler-/Beobachtersauslegung bis hin zur Inbetriebnahme adressieren und im Folgenden genauer darauf eingehen, mit welchen Methoden dies umgesetzt werden soll. Ein weiteres wesentliches Ziel ist die Entwicklung von Handlungsempfehlungen, die interessierten Leser\*innen konkrete Hinweise zur effektiven Anwendung der entwickelten Methoden geben.

## 1.2 Forschungsumfeld der Nachwuchsgruppe

An der Universität Paderborn und dem Heinz Nixdorf Institut wird schon seit längerer Zeit großer Wert auf Forschung im Kontext von intelligenten technischen

Systemen gelegt. Daher muss auch die Entstehung der Forschungsgruppe DART in diesem Kontext gesehen werden, da sie von den vorangehenden Entwicklungen beeinflusst wurde und von der kontinuierlichen Unterstützung in diesem Bereich profitiert hat. Aus diesem Grund werden an dieser Stelle ehemalige und aktuelle Forschungsprojekte und Institutionen vorgestellt, die vom Team der Nachwuchsgruppe als wichtig und einflussreich angesehen werden.

Zuerst ist das Heinz Nixdorf Institut HNI zu nennen, dem die Nachwuchsgruppe DART angehört. Das Leitbild des HNI beschäftigt sich mit intelligenten technischen Systemen für die Märkte von morgen. Dabei steht die Symbiose von Informatik und Ingenieurwissenschaften im Zentrum der Arbeiten. Daraus ergeben sich Impulse für intelligente technische Systeme und entsprechende Dienstleistungen. Die Forschung soll dazu beitragen, neue Arbeitsplätze zu schaffen, den Wohlstand zu erhalten und die nachhaltige Entwicklung zu fördern. Um diese Ziele zu erreichen, besitzt das HNI verschiedene Forschungskompetenzen und Anwendungsbereiche. Das Projekt DART findet sich dabei im Bereich der intelligenten technischen Systeme wieder, welcher sich mit den Themen Lernfähigkeit, Adaptivität und Regelung beschäftigt. Hier werden Lösungen etabliert, die es ermöglichen, dass technische Systeme selbst in hochkomplexen, sich dynamisch verändernden Umgebungen funktionieren, vielfach ohne zentrale Koordination. Ermöglicht wird dies durch die Kombination klassischer Regelungs- und Steuerungstechnik mit modernen Methoden der künstlichen Intelligenz und des maschinellen Lernens, die solchen Systemen ein hohes Maß an Autonomie verleihen und mit der Fähigkeit der datengetriebenen Anpassung ihres Verhaltens ausstatten. Damit stimmen die Forschungsziele des HNI in diesem Bereich vollständig mit der Grundidee des Projektes DART überein und die Zusammenarbeit und Kooperation im Institut unterstützte den Gesamterfolg der Nachwuchsgruppe<sup>1</sup>.

Auch weitere Forschende der Universität Paderborn beschäftigen sich schon seit Jahren mit dem Thema der intelligenten technischen Systeme, was in einem eigenen Profilbereich „Intelligente technische Systeme“ zusammengefasst ist, um die Forschungsideen zu bündeln und zu koordinieren. Hierbei wird betont, dass intelligente technische Systeme durch das Zusammenspiel von Algorithmen, Informationstechnik, Mechanik, Sensorik und Aktorik gekennzeichnet sind. Diese Komponenten bzw. Teilsysteme sind miteinander vernetzt und können auf verschiedenen räumlichen und zeitlichen Skalen operieren. Beispiele für intelligente Systeme, die auch an der Universität Paderborn als Anwendungsfelder bearbeitet werden, sind Smart Grids, die digitale Fabrik, Erklärbare KI, Smart Cities, autonome Fahrzeuge oder autonome Roboter. Eine Integration dieser Systeme in einen Anwendungskontext bedeutet hohe Anforderungen im Hinblick auf Sicherheit, Robustheit, Lernfähigkeit, Ressourceneffizienz, Datenschutz u.a. Aufgrund der Komplexität und Heterogenität der Systeme sind diese Anforderungen nicht leicht zu erfüllen. Folglich ist ein interdisziplinärer Forschungsansatz unbedingt erforderlich, weil sich das Verhalten des Gesamtsystems erst aus dem komplexen Zusammenwirken der Komponenten, der Interaktion mit den Nutzer\*innen und den Besonderheiten der Anwendung ergibt. Die Forschungsarbeiten im Profilbereich intelligente technische Systeme sind interdisziplinär aufgestellt und befassen sich mit Analyse und Entwurf (Modell-

---

<sup>1</sup><https://www.hni.uni-paderborn.de/nachwuchsgruppe-dart>

bildung, Simulation, Verifikation, Methodenentwicklung) von intelligenten technischen Systemen und schließen sozioökonomische und kulturwissenschaftliche Aspekte (Geschäftsmodelle, Verhaltensökonomie, Technikethik, Interaktionsdesign) ein. Durch diese Charakterisierung des Profilbereichs wird deutlich, dass sich auch hier die Forschungsthemen von DART direkt einbetten lassen und eine solch intensive Beschäftigung vieler Forscher an der Universität Paderborn als sehr produktiv angesehen werden kann<sup>2</sup>.

Eine weitere Institution, das im Rahmen der Forschungsarbeit von DART genannt werden muss, ist das Fraunhofer Institut für Entwurfstechnik Mechatronik IEM<sup>3</sup>. Auch das IEM verschreibt sich u.a. der Erforschung intelligenter technischer Systeme, wodurch deutlich wird, wie zentral dieses Gebiet am Gesamtstandort Paderborn ist. Das IEM sieht das Fundament erfolgreicher technischer Systeme in einer ganzheitlichen Entwicklung. Das Institut unterstützt Unternehmen jeder Größe, fachübergreifendes Engineering in die Praxis zu bringen. Von der maßgeschneiderten Produktentwicklung, über langfristige Konzepte für ein modellbasiertes und sicheres Engineering hin zur Umsetzung digitaler Zukunftsvisionen wie Industrie 4.0 basiert die Arbeit des Fraunhofer IEM auf sechs Kernkompetenzen: Intelligente Technische Systeme, Digitale Transformation, Systems Engineering, Virtualisierung und Modellbildung, IT-Security und Software Engineering. Es wird festgestellt, dass intelligente technische Systeme eine breite Basis hoch anspruchsvoller Technologien benötigt: Ob virtuelle Sensorik, selbstoptimierende Regelungen, maschinelles Lernen oder Augmented und Virtual Reality, das Fraunhofer IEM beherrscht den Stand der Technik und setzt in einzelnen Gebieten Standards. Dabei steht die bedarfsorientierte Anwendung der Technologien im Mittelpunkt. Und aus diesem Grund ist die Kooperation der Nachwuchsgruppe DART mit dem IEM so spannend: Der gegenseitige Austausch informiert über die Probleme und Anwendungsfelder, die in Industrieunternehmen von Interesse sind.

Historisch gesehen konnte sich das Forschungsthema der Nachwuchsgruppe aus zwei großen Forschungsprojekten, die aufeinander aufbauten, entwickeln: Dem Sonderforschungsbereich SFB 614 *Selbstoptimierende Systeme des Maschinenbaus* und dem darauf zeitlich direkt folgende Spitzenclusterprojekt *it's OWL Intelligente Technische Systeme OstWestfalenLippe*.

Der SFB 614, der von 2002 bis 2013 an der Universität Paderborn lief, war ein zentrales Element in der Erforschung intelligenter technischer Systeme. In diesem Großprojekt wurde erkannt, dass sich aus der zunehmenden Durchdringung des Maschinenbaus mit Informationstechnik erhebliche Erfolgspotentiale eröffnen. Der Begriff Mechatronik bringt dies zum Ausdruck – gemeint ist damit das enge Zusammenwirken von Mechanik, Elektronik, Regelungstechnik und Softwaretechnik. Moderner Maschinenbau ist Mechatronik. Künftige Systeme des Maschinenbaus werden aus Konfigurationen von Systemelementen mit einer inhärenten Teilintelligenz bestehen. Das Verhalten des Gesamtsystems wird durch die Kommunikation und Kooperation der intelligenten Systemelemente geprägt sein. Aus informationstechnischer Sicht handelt es sich nach dem Verständnis der Forschergruppe des SFB um verteilte Systeme von miteinander kooperierenden Agenten. Daraus eröffnen sich

---

<sup>2</sup><https://www.uni-paderborn.de/forschung/forschung-im-profil/intelligente-technische-systeme>

<sup>3</sup><https://www.iem.fraunhofer.de>

faszinierende Möglichkeiten für die Gestaltung der maschinenbaulichen Erzeugnisse von morgen. Der Begriff Selbstoptimierung charakterisiert diese Perspektive: Selbstoptimierung ermöglicht handlungsfähige Systeme mit inhärenter „Intelligenz“, die in der Lage sind, selbstständig und flexibel auf veränderte Umgebungsbedingungen zu reagieren<sup>4</sup>.

Die Fortführung der Themen des SFB 614 konnten durch die erfolgreiche Beantragung des Spitzenclusters it's OWL weitergeführt werden. Inzwischen ist hieraus ein großes Technologie-Netzwerk mit dem Thema intelligente technische Systeme entstanden, in dem über 200 Unternehmen, Forschungseinrichtungen und Organisationen Lösungen für intelligente Produkte und Produktionsverfahren entwickeln. Dabei steht it's OWL für Innovationsgeist, Technologiekompetenz, Forschungsexpertise und gelebten Technologietransfer. Kennzeichen von it's OWL sind das enge Zusammenspiel von mittelständischen Technologieführern und anwendungsorientierten Forschungseinrichtungen im produzierenden Gewerbe. Neue Technologien werden gemeinsam erschlossen und in praxisnahen Angeboten für kleine und mittlere Unternehmen zur Verfügung gestellt. Seit 2022 hat sich it's OWL zum Ziel gesetzt, Ostwestfalen Lippe zur Modellregion für nachhaltige Wertschöpfung zu entwickeln<sup>5</sup>.

Damit sind die Arbeiten und auch die Ergebnisse der Nachwuchsgruppe DART nicht als alleinstehend zu betrachten, sondern stehen in einer Linie zu vorausgehenden und zukünftigen Projekten im Bereich der intelligenten Systeme an der Universität Paderborn und ihrem Forschungsumfeld in der Region. Der Beitrag der Nachwuchsgruppe liegt dabei in der Erforschung von grundlegenden Methoden, die maschinelles Lernen mit regelungstechnischen Ansätzen synergetisch verbinden.

### 1.3 Motivation für datengetriebene Methoden in der Regelungstechnik

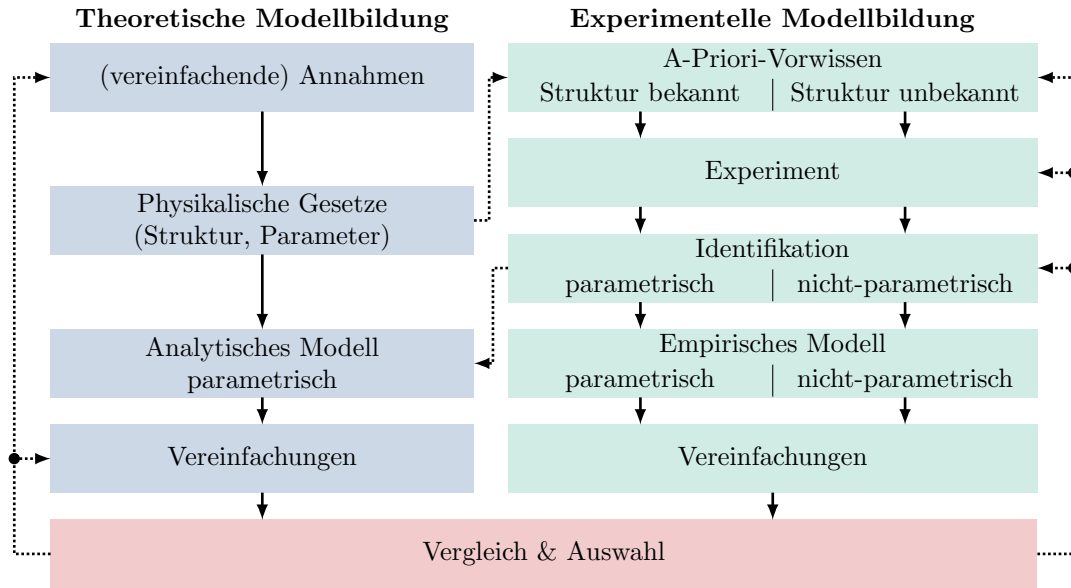
Der Begriff KI – Künstliche Intelligenz wird mit den immer größer werdenden Rechenkapazitäten seit einigen Jahren sehr gehypt. Dieses Phänomen ist im Alltag beispielsweise durch Diskussionen zum Thema „Was sind Chancen und Gefahren der KI?“ sichtbar und auch viele Forschungsgruppen beschäftigen sich mit den Möglichkeiten der KI in ihren vielfältigen Anwendungen. Die Verwendung von Methoden, die mit Daten arbeiten ist dagegen in der Regelungstechnik schon seit längerer Zeit etabliert, wobei hier nicht einmal KI oder maschinelles Lernen gemeint sein muss. Für jede Regelung eines technischen System sind wir darauf angewiesen, Messdaten zurückzuführen und diese innerhalb des Regelungsansatzes auszuwerten. Ansonsten wäre es nicht möglich, die Grundidee einer Regelung umzusetzen. Daher ist es im Kontext der Regelungstechnik nur natürlich darüber nachzudenken, ob die vorhandenen gemessenen Daten auch zur zusätzlichen Weiterverarbeitung genutzt werden können, um möglicherweise eine höhere Form des Wissens über das System zu erhalten. Dies kann auch mit Lernen bezeichnet werden. Ebenso bei der Modellierung von technischen Systemen wird die klassische Verwendung von Daten deutlich: Sie kann sowohl durch eine theoretische als auch durch eine experimentelle

---

<sup>4</sup><https://www.uni-paderborn.de/projekt/281>

<sup>5</sup><https://www.its-owl.de/home>



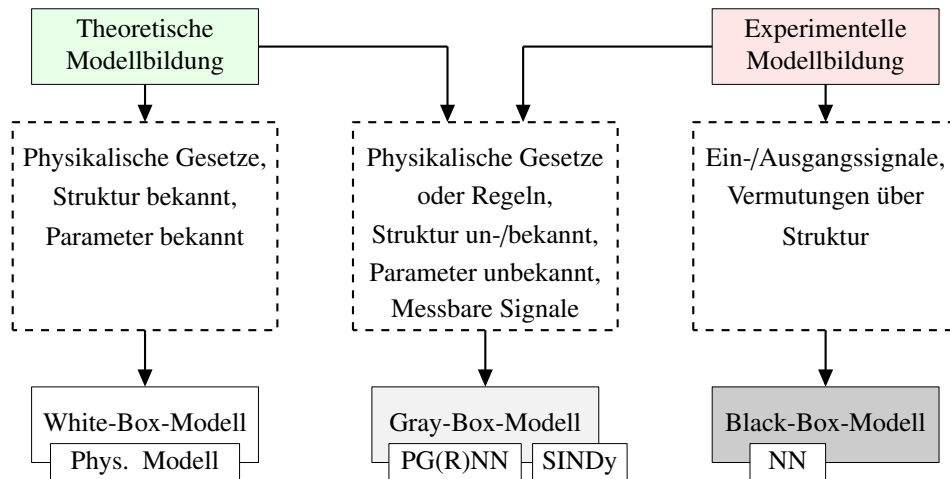


**Abbildung 1.3** Gegenüberstellung der unterschiedlichen Vorgehensweisen bei der theoretischen und experimentellen Modellbildung. In Anlehnung an [1].

Modellbildung erfolgen, außerdem gibt es Mischformen. Diese sind nach [1] in der Abbildung 1.3 dargestellt und unterscheiden sich darin, ob Vorwissen aus physikalischen Gesetzen oder Messdaten aus Experimenten zur Identifikation eines Modells genutzt werden. Das Vorgehen der beiden Modellierungsstrategien ähnelt sich an einigen Stellen. Beide entwickeln ausgehen von Vorwissen und Annahmen ein Modell, welches anschließend ggf. noch vereinfacht wird. Zudem ergänzen sich die Strategien unter Umständen durch Teilelemente untereinander, z.B. bei der Parameteridentifikation des analytisch entwickelten Modells durch Messdaten.

Daher sind die Grenzen zwischen den Perspektiven der physikalischen und experimentellen Modellbildung fließend zu sehen und nicht unbedingt streng definiert, da viele Modelle sowohl mittels physikalischer Gesetzmäßigkeiten als auch durch Messdaten beschrieben werden können. Diese können daher je nach Art der verwendeten Informationen und Kenntnis der Systemstruktur als White-, Gray-, oder Black-Box-Modelle charakterisiert werden. In der Abbildung 1.4 ist diese Unterteilung dargestellt: Ein White-Box-Modell liegt vor, wenn die Systemstruktur sowie Parameter durch physikalische Gesetzmäßigkeiten bekannt sind und damit ein physikalisch motiviertes Modell aufgestellt werden kann. Black-Box-Modelle stellen dagegen das Ein-/Ausgangsverhalten dar und erlauben lediglich Vermutungen zur Struktur. Ein Beispiel für diese Art von Modellen sind neuronale Netze, die keinerlei physikalische Interpretation zulassen. Modelle, die sowohl aus der theoretischen als auch aus der experimentellen Modellbildung hervorgehen, werden als Gray-Box-Modelle bezeichnet. Diese Modelle, deren Anteil von theoretischer bzw. experimenteller Modellbildung variieren kann, werden im weiteren Verlauf auch als *hybrid* bezeichnet. Zu ihnen werden häufig auch Modelle gezählt, die ML-Methoden sinnvoll mit regelungstechnischen Ansätzen verbinden, beispielsweise die Methoden PGNN in Abschnitt 3.3 oder SINDy in Abschnitt 3.2.

Die Arbeit mit Methoden des maschinellen Lernens, also mit Methoden, die Beispieldaten eines Systems auf ein mathematisches Modell abbilden, kann somit prinzi-



**Abbildung 1.4** Modellkategorien basierend auf dem Grad der verwendeten Messdaten und des physikalischen Vorwissens, angelehnt an [1].

piell relativ einfach auch für regelungstechnische Systeme durchgeführt werden. Dabei haben wir nie den Anspruch Systeme zu erzeugen, die KI besitzen, denn dies wird im Allgemeinen mit der Nachahmung von menschlichem Verhalten assoziiert. Wir wollen ML nutzen, um die vorhandenen oder einfach erhaltenen Daten vorteilhaft zu nutzen und zum Beispiel Modelle oder Regelungsansätze zu verbessern sowie adaptiv gestalten zu können. Dabei wollen wir aber weiterhin das System sehr gut verstehen können, was besonders gut mit einer physikalisch motivierten Beschreibung in Form von Differentialgleichungen funktioniert. Daher ist eines der Ziele der Nachwuchsgruppe DART die Interpretierbarkeit von hybriden Modellen bestehend aus klassischer Regelungstechnik und datengetriebenen Methoden zu gewährleisten und die Verwendung von Black-Box-Modellen wie beispielsweise neuronalen Netzen zu vermeiden. Neben dem Wunsch der Interpretierbarkeit gibt es weitere Aspekte, die in der hybriden Modellierung von Modellen und Regelungsansätzen berücksichtigt werden sollen. Hierzu gehören

- ein geringer Modellierungsaufwand,
- ein geringer Rechenaufwand bei der Prädiktion,
- eine hohe Dateneffizienz,
- die physikalische Interpretierbarkeit,
- die Nutzung von physikalischem Vorwissen,
- die Anwendung bestehender Entwurfsverfahren,
- die Adaptionfähigkeit bei Systemveränderungen,
- gutes Extrapolationsverhalten.

Diese Punkte sind wichtig, damit sich neu entwickelte hybride Ansätze gut in das Vorgehen in der Regelungstechnik einbinden lassen. Sie werden in Abschnitt Abschnitt 2.3 genauer erläutert und innerhalb dieser Monografie wieder aufgegriffen,

### *1.3 Motivation für datengetriebene Methoden in der Regelungstechnik*

um die entwickelten Methoden einzuordnen. Abschließend lässt sich sagen, dass es viele Gründe für die Nutzung von datengetriebenen Ansätzen im Bereich der Regelungstechnik gibt und sich diese sehr gut anhand von mechatronischen Systemen motivieren lassen. Die Forschungsergebnisse in diesem Beitrag werden zeigen, dass vor allem Methoden mit ungenauen oder fehlerhaften Modellen und adaptive Ansätze davon profitieren, dass kontinuierlich Daten von technischen Systemen gesammelt werden und diese vorteilhaft genutzt werden können. Im Folgenden werden entlang des regelungstechnischen Entwurfsprozesses die durch die Nachwuchsgruppe DART entwickelten Methoden vorgestellt und in den Stand der Technik eingebettet.



## 2 Grundlagen des maschinellen Lernens in der Regelungstechnik

Methoden des maschinellen Lernens werden inzwischen in vielen wissenschaftlichen Fächern verwendet und daraufhin analysiert, ob sie etablierte Verfahren verbessern können. Die Nachwuchsgruppe DART hat ebendies für die Anwendung auf regelungstechnische Probleme untersucht, daher sollen an dieser Stelle einige Grundlagen erläutert werden, die für das Verständnis von ML und die Anwendung auf regelungstechnische Methoden wichtig sind. Dafür motivieren wird in Abschnitt 2.1 zunächst motiviert, warum ausgewählte ML-Ansätze vorteilhaft in der Regelungstechnik eingesetzt werden können. Anschließend werden in Abschnitt 2.2 grundlegende Begriffe und Eigenschaften von ML-Ansätzen allgemein erläutert, sodass diese im Verlauf dieses Beitrages vorausgesetzt werden können. Einzelne grundlegende ML Methoden, die im Rahmen unserer Arbeit verwendet oder weiterentwickelt werden, werden zusätzlich an den entsprechenden Stellen in den folgenden Kapiteln eingeführt. Dazu gehören beispielsweise neuronale Netze (siehe Unterabschnitt 3.3.1) oder die Methode der Bayesschen Optimierung (siehe Unterabschnitt 4.4.3). Abschließend für dieses grundlegende Kapitel werden Kriterien und Ziele formuliert, die sich bei der Entwicklung von regelungstechnisch wertvollen hybriden Methoden als wichtig herausgestellt haben und in den neu entwickelten Ansätzen der Nachwuchsgruppe DART Berücksichtigung finden.

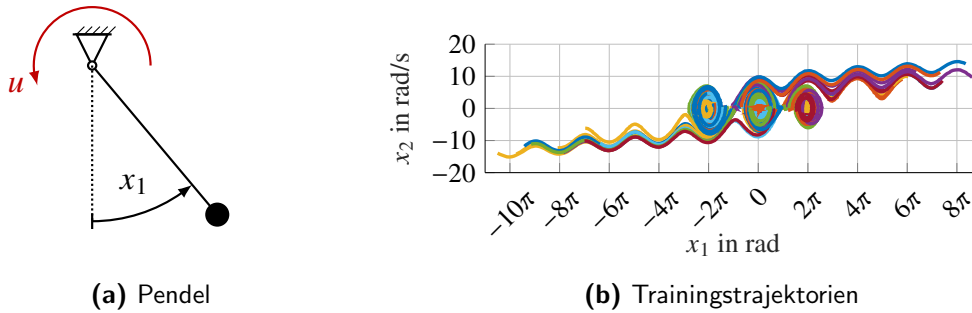
### 2.1 Datengetriebene Methoden

Beim Aufbau von hybriden Methoden stellt sich die Frage, welche Methoden des maschinellen Lernens sinnvollerweise eingesetzt werden sollten. Dies wird an dieser Stelle durch zwei Methoden motiviert, die sich in ihrem Lernansatz grundsätzlich unterscheiden: Einem (flachen) neuronalen Netz (siehe Abschnitt 3.3) im Vergleich zu der physikalisch motivierten Bibliotheksmethode SINDy (siehe Abschnitt 3.2). Dazu betrachten wir ein nichtlineares Pendel, welches im Gelenk aktuiert ist und in der Abbildung 2.1(a) dargestellt wird. Der Zustandsvektor lautet  $\mathbf{x} = [x_1, x_2]^\top = [\varphi, \dot{\varphi}]^\top$  und die Differentialgleichungen ergeben sich zu

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} x_2 \\ -\frac{g}{l} \sin x_1 - \frac{d}{ml^2} x_2 + \frac{1}{ml^2} u \end{bmatrix}, \quad (2.1)$$
$$m = 1\text{kg}, g = 9,81 \frac{\text{m}}{\text{s}^2}, l = 0,5\text{m}, d = 0,1 \frac{\text{kg m}^2}{\text{s}}.$$

Es wird angenommen, dass der Zustandsvektor  $\mathbf{x}$  vollständig messbar ist. Außerdem besitzt das System für  $u = 0$  eine stabile untere Ruhelage in  $\mathbf{x}_1^* = [2k\pi, 0]^\top$  und eine instabile obere Ruhelage in  $\mathbf{x}_2^* = [(2k+1)\pi, 0]^\top$  mit  $k \in \mathbb{Z}$ . An diesem Beispiel

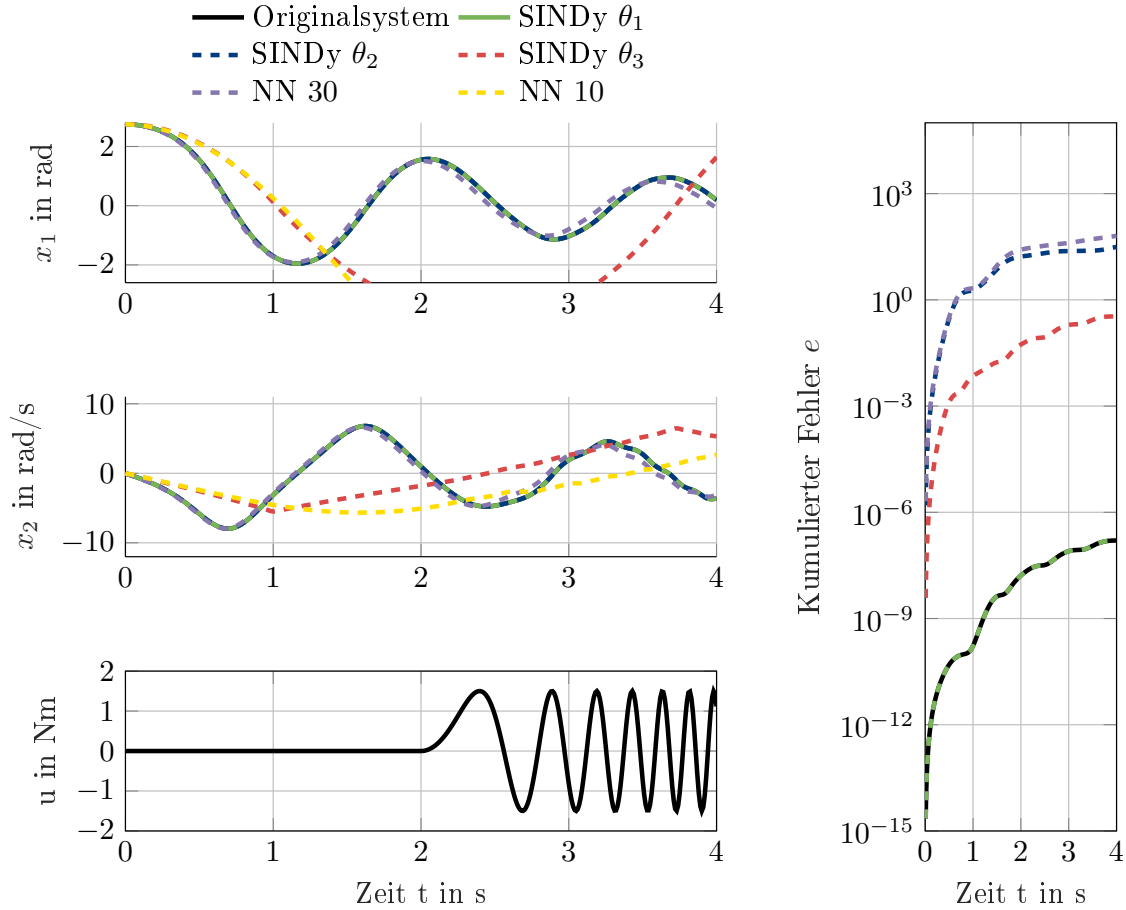
mit einfachem dynamischen Verhalten soll nun gezeigt werden, wie mit relativ wenig Vorwissen die Dynamik mittels unterschiedlicher ML Methoden gelernt werden kann. Das Vorwissen sind hierbei nicht die oben beschriebenen Differentialgleichungen, sondern wir gehen ausschließlich davon aus, dass wir wissen, dass das System durch zwei Zustände beschrieben werden kann, eine Eingangsgröße im Gelenk besitzt und dass Messgrößen der Zustände und des Eingangs erzeugt werden können. Hierfür bereiten wir diese Messdaten, die anschließend die Trainingsdaten für die Lernalgorithmen darstellen, vor, indem 100 Simulationen (jeweils mit einer Dauer von 3 s und einer Schrittweite von  $\Delta t = 0.01$  s) mit zufälligen Anfangsauslenkungen  $\mathbf{x}_0$  im Zustandsraum und zufälligen Systemeingängen  $u$  durchgeführt werden vgl. Abbildung 2.1(b). Die Messeinrichtung wurde durch ein additives weißes gaußsches Rauschen mit einer Standardabweichung von  $\sigma = 0.02$  modelliert. Die Testtrajektorie ist so gewählt, dass das Pendel zunächst autonom aus der Anfangsauslenkung  $\mathbf{x}_0 = [\frac{7\pi}{8}, 0]^T$  schwingt und nach 2 s sinusförmig mit ansteigender Schwingungsfrequenz angeregt wird. Dadurch erfährt das Pendel einerseits den Einfluss der Stellgröße und andererseits, aufgrund der hohen Auslenkung, die nichtlinearen Anteile der Dynamik.



**Abbildung 2.1** Das nichtlineare aktuierte Pendel illustriert als einfaches Beispiel die unterschiedlichen Strategien zur Modellbildung in der Regelungstechnik.

Die Dynamik des Pendels lässt sich schon durch ein flaches neuronales Netz vergleichsweise gut approximieren. Dabei ist mit einem flachen neuronalen Netz ein mehrschichtiges Perzeptron gemeint, das nur eine versteckte Schicht besitzt im Gegensatz zu tiefen neuronalen Netzen, die hunderte versteckte Schichten besitzen können, siehe [2]. Das Netz bildet dabei die zeitdiskrete Dynamik ab und besitzt daher drei Eingänge  $\mathbf{in} = [x_{1,k}, x_{2,k}, u_k]^T$  und zwei Ausgänge  $\mathbf{out} = [x_{1,k+1}, x_{2,k+1}]^T$ . Die versteckte Schicht besteht aus 30 Neuronen, die mit einer hyperbolischen Aktivierungsfunktion  $f_{act}(x) = \frac{2}{1+e^{-2x}} - 1$  basierend auf dem Tangens betrieben werden. Freie Parameter im neuronalen Netz sind die Gewichte auf den Verbindungen zwischen verschiedenen Neuronen und ein Bias in jedem Neuron, sodass das hier aufgestellte Netz auf  $3 \cdot 30 + 30 = 120$  Parameter zwischen der Eingangs- und versteckten Schicht und  $30 \cdot 2 + 2 = 62$  Parameter zwischen der versteckten Schicht und der Ausgangsschicht kommt. Insgesamt ergeben sich 182 freie Parameter, die durch einen Lernalgorithmus basierend auf Optimierungsmethoden angepasst werden.

Das Ergebnis des Lernens durch ein flaches neuronales Netz mit 30 Neuronen ist in der Abbildung 2.2 dargestellt. Es wird deutlich, dass das aufgestellte neuronale Netz in der Lage ist, die Dynamik der Testtrajektorie mit einer hohen Genauig-



**Abbildung 2.2** Vergleich der Approximation der Pendeldynamik mit unterschiedlichen neuronalen Netzen mit 30 bzw. 10 versteckten Neuronen und der bibliotheksbasierten Methode SINDy mit unterschiedlichen Bibliotheksansätzen (siehe Gleichung (2.2)).

keit abzubilden und dadurch das Lernziel grundsätzlich erfüllt werden konnte. Weiterführend muss aber überlegt werden, wie mit dieser Struktur weiter umgegangen werden kann. Das neuronale Netz ist eine Black-Box Struktur (siehe Abbildung 1.4) und kann daher ausschließlich das Ein-/ Ausgangsverhalten abbilden. Wenn sich einer der Systemparameter, beispielsweise der Dämpfungskoeffizient  $d$ , ändern würde, dann müsste der Lernprozess inklusive der Gewinnung der Trainingsdaten komplett von vorne begonnen werden. Außerdem ist das neuronale Netz nicht in der Lage die zu Grunde liegenden physikalischen Terme zu identifizieren. Somit wird beispielsweise nicht klar, dass die Dynamik des Pendels durch die Sinusfunktion schwingungsfähig ist (siehe Gleichung (2.1)). Eine Regelung könnte nur durchgeführt werden, wenn die Kenntnis des Ein-/Ausgangsverhaltens ausreichend hierfür ist und Stabilitätsuntersuchungen basierend auf der Analyse von mathematischen Gleichungen können nicht durchgeführt werden. Daher kann festgehalten werden, dass ein neuronales Netz prinzipiell sehr gut in der Lage ist mit relativ wenig Aufwand die Dynamik eines technischen Systems zu erlernen, aber je nach geplanter zukünftiger Anwendung dieses Modells kann es nicht geeignet weiterverwendet werden.

In einem zweiten Schritt ist auch ein neuronales Netz mit 10 Neuronen in der versteckten Schicht trainiert worden. Das Ergebnis ist ebenfalls in der Abbildung 2.2 zu

sehen und zeigt, dass die Komplexität dieses neuronale Netz nicht ausreicht, um die Dynamik des einfachen Pendels korrekt abzubilden. Schon nach einer kurzen Zeit von ca. 0,3 s weicht das approximierte Systemverhalten deutlich vom echten System ab. Dadurch wird deutlich, dass ein neuronales Netz immer gut auf das vorliegende System angepasst werden muss. Eine zu hohe oder zu geringe Komplexität des Modells, hier ausgedrückt durch die Anzahl der versteckten Neuronen und somit der freien Parameter im System, wirkt sich negativ auf das Lernergebnis aus.

Im Vergleich zu neuronalen Netzen, die ein Black-Box-Modell der Dynamik erzeugen, versucht die Methode SINDy (siehe Abschnitt 3.2) physikalische Terme durch die Auswahl aus einer Funktionsbibliothek zu identifizieren, die in der Lage sind die Dynamik zu beschreiben. Damit ist diese Methode den Gray-Box-Methoden zuzuordnen. SINDy basiert dabei auf der Idee, dass technische Systeme in der Regel durch eine begrenzte Zahl an physikalischen Termen zu beschreiben sind und beschränkt daher die Auswahl von physikalischen Termen aus der zu Grunde liegenden Bibliothek auf eine möglichst kleine Anzahl. Um SINDy anzuwenden, muss zunächst eine Bibliothek erstellt werden, aus der die Beschreibung extrahiert werden soll. Dabei sollte nach Möglichkeit Wissen über das dynamische Verhalten des Systems durch die Nutzung in der Bibliothek eingebracht werden. Im vorliegenden Beispiel werden drei unterschiedliche Bibliotheken auf ihre Eignung zum Erlernen der Pendeldynamik untersucht:

$$\begin{aligned}\Psi_1(\mathbf{x}, u) &= [x_1 \quad x_2 \quad \sin x_1 \quad u]^T, \\ \Psi_2(\mathbf{x}, u) &= [x_1 \quad x_2 \quad \sin x_1 \quad \sin x_2 \quad \cos x_1 \quad \cos x_2 \quad x_1^2 \quad x_2^2 \quad u]^T, \\ \Psi_3(\mathbf{x}, u) &= [x_1 \quad x_2 \quad \text{sign } x_1 \quad \text{sign } x_2 \quad x_1^2 \quad x_2^2 \quad u]^T.\end{aligned}\quad (2.2)$$

Es wird deutlich, dass die Bibliothek  $\Psi_1$  alle Terme enthält, um die Differentialgleichungen des Pendels korrekt anzunähern. Bei  $\Psi_2$  sind diese Terme ebenfalls enthalten, aber es werden zusätzlich weitere zur Auswahl angeboten. Die dritte Bibliothek  $\Psi_3$  beinhaltet dann nicht mehr den Sinusterm, sondern dynamisch eher unpassende Anteile wie  $\text{sign } x_1$ . Mit Hilfe der gleichen Trainingsdaten wie im Fall des neuronalen Netzes bestimmt nun der Algorithmus von SINDy die Parameter, die für die Systembeschreibung am besten passend sind, so dass sich beispielsweise eine Systembeschreibung durch

$$\begin{aligned}\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} &= \boldsymbol{\theta}_1 \cdot \Psi_1 = \begin{bmatrix} 0 & 1.0000 & 0 & 0 \\ 0 & -0.4000 & -19.6203 & 4.0000 \end{bmatrix} \cdot \begin{bmatrix} x_1 \\ x_2 \\ \sin x_1 \\ u \end{bmatrix} \\ &= \begin{bmatrix} x_1 \\ -0.4x_2 - 19.6203 \sin x_1 + 4u \end{bmatrix}\end{aligned}\quad (2.3)$$

bei der Verwendung der Bibliothek  $\Psi_1$  ergibt. Die vollständigen Parametermatrizen lauten

$$\boldsymbol{\theta}_1 = \begin{bmatrix} 0 & 1.0000 & 0 & 0 \\ 0 & -0.4000 & -19.6203 & 4.0000 \end{bmatrix},$$



$$\boldsymbol{\theta}_2 = \begin{bmatrix} 0 & 1.0000 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -0.4000 & -19.6203 & 0 & 0 & 0 & 0 & 4.0000 \end{bmatrix},$$

$$\boldsymbol{\theta}_3 = \begin{bmatrix} 0 & 1.0000 & 0 & 0 & 0 & 0 & 0 \\ 0.3799 & 0.0309 & -5.3819 & 0.6962 & 0.0057 & -0.0094 & 0.9197 \end{bmatrix}.$$

und müssen immer in Kombination mit den jeweiligen Bibliotheken  $\Psi_i, i = 1, \dots, 3$  gesehen werden.

Bei Betrachtung der Parametermatrizen  $\boldsymbol{\theta}_1$  und  $\boldsymbol{\theta}_2$  wird deutlich, dass sich durch die Anwendung von SINDy der korrekte Parametersatz, der aus den DGLn aus Gleichung (2.1) bekannt ist, ergibt. Das heißt, dass SINDy in der Lage ist, die Dynamik des Pendels mit Hilfe der Vorgabe einer passenden physikalisch motivierten Bibliothek vollständig abzubilden. Dieses Resultat sieht man auch in Abbildung 2.2, in der die drei Ergebnisse von SINDy ebenfalls dargestellt werden. Sowohl der Winkel  $\varphi$  als auch die Winkelgeschwindigkeit  $\dot{\varphi}$  zeigen das gleiche Verhalten, wie das Originalsystem. Der geringe Fehler wird ausschließlich durch die simulierten Rauschprozesse verursacht. Wenn dagegen die physikalisch eher weniger gut passende Bibliothek  $\boldsymbol{\theta}_3$  verwendet wird, ergibt sich ein ähnlich schlechtes Ergebnis wie bei dem neuronalen Netz mit nur 10 versteckten Neuronen und die Prädiktion stimmt nur für einen sehr kurzen ersten Teil der Trajektorie mit der echten Dynamik des Systems überein.

Diese Analyse zeigt somit, dass bei beiden Ansätze, also dem flachen neuronalen Netz und auch SINDy sowohl sehr gute als auch schlechte Ergebnisse produzieren können, je nachdem, ob die die Methode beeinflussenden Parameter gut oder schlecht gewählt worden sind. Diese Aspekte, wie die Anzahl von Neuronen oder die Auswahl der Bibliothek, werden im Bereich des ML auch als Hyperparameter bezeichnet und charakterisieren den Erfolg von Methoden entscheidend. Ein Vorteil der physikalisch motivierten Methode SINDy ist es, dass die Testung von unterschiedlichen Bibliotheken vergleichsweise einfach verstanden werden kann. Offensichtlich spielen viele Terme der Bibliothek  $\boldsymbol{\theta}_2$  keine Rolle in der Dynamik des Pendels und bei Betrachtung von  $\boldsymbol{\theta}_3$  wird deutlich, dass der Algorithmus keine dünnbesetzte Struktur bestimmen konnte und damit der Schluss nahe liegt, dass die betrachteten physikalischen Terme nicht für die Beschreibung der Dynamik geeignet sind.

Aus diesem Beispiel ziehen wir einige Konsequenzen für unsere Arbeit mit Methoden, die aus klassischer Regelungstechnik und Methoden des maschinellen Lernens bestehen und somit hybrid sind. Es ist vorteilhaft, bekanntes **physikalisches Wissen** zu verwenden und in die hybride Methode einbringen zu können, da dadurch auch die Komplexität des ML Modells weniger gering sein kann. Außerdem streben wir an, dass die Ergebnisse des Algorithmus **physikalisch interpretierbar** sind und somit von der anwendenden Person im ingenieurwissenschaftlichen Kontext auch gut verstanden werden können.

Neben diesen zwei Aspekten gibt es im Kontext hybrider regelungstechnischer Systeme auch weitere Kriterien und Ziele, die bei einer Entwicklung berücksichtigt werden sollten. Auf diese wird ausführlich in Abschnitt 2.3 eingegangen.

## 2.2 Grundlagen des maschinellen Lernens

Je nach Sichtweise existieren zahlreiche Definitionen für den Begriff der künstlichen Intelligenz (KI), wobei sich die grundlegende Idee folgendermaßen charakterisieren lässt:

”Künstliche Intelligenz ist die Eigenschaft eines IT-Systems, menschenähnliche, intelligente Verhaltensweisen zu zeigen.”[3]

Die Intelligenz einer KI lässt sich durch die Fähigkeit des Lernens charakterisieren. Lernen meint hier, dass die KI künstliches Wissen aus Erfahrung generieren kann, d. h. anhand von Beispielen (Lerndaten) werden Muster erkannt und verallgemeinert mit dem Ziel nach der Lernphase auch bisher unbekannte Daten verarbeiten zu können. Im ingenieurwissenschaftlichen Bereich können Algorithmen mit der Fähigkeit zu Lernen die Automatisierung komplexer Aufgaben erleichtern, indem gewisse menschliche Wahrnehmungs- und Verstandsleistungen durch maschinelles Lernen (ML) realisiert werden. Der Begriff des maschinellen Lernens lässt sich folgendermaßen definieren:

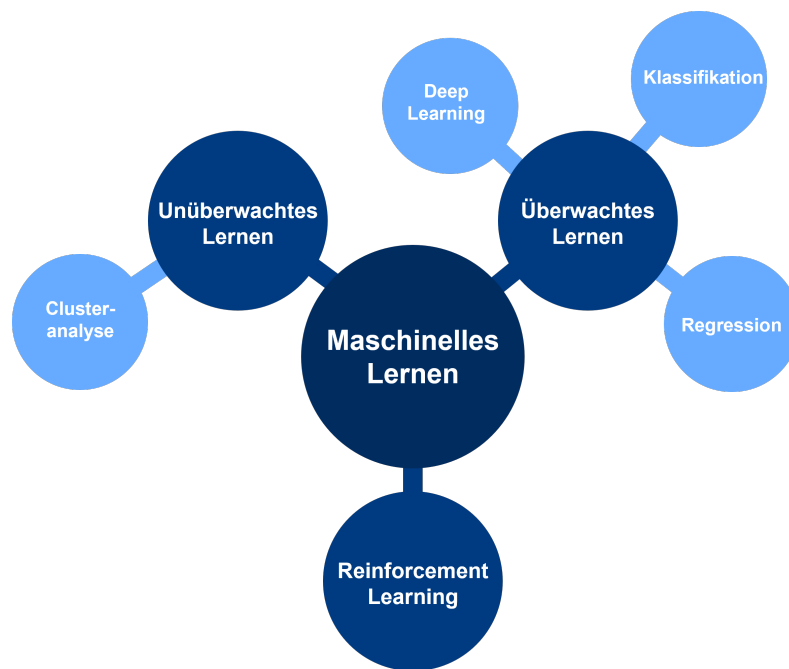
”ML bezeichnet Verfahren, bei denen Computer-Algorithmen aus Daten lernen, beispielsweise Muster zu erkennen oder gewünschte Verhaltensweisen zu zeigen, ohne dass jeder Einzelfall explizit programmiert wurde. [...] Oft wird Maschinelles Lernen mit KI gleichgesetzt. Während in der KI häufig ML eingesetzt wird, ist ML eine Methode, ein Werkzeug unter vielen der KI.”[3]

In anderen Quellen wird eine Unterscheidung für starke und schwache KI vorgeschlagen (siehe zum Beispiel [4]), welche ähnlich unterschieden werden wie die gerade betrachteten Klassen KI und ML. Starke KI-Systeme sind Maschinen mit menschenähnlicher Intelligenz oder ihnen sogar überlegen und die sich selbst unabhängig vom Menschen außerhalb ihres ursprünglichen Anwendungszwecks weiterentwickeln. Bis heute sind keine Ansätze für starke KI bekannt und ob sie in der Zukunft existieren werden, kann aktuell nicht vorhergesagt werden. Dagegen beschreibt die schwache KI Anwendungen, die von Ingenieur\*innen speziell auf einen Anwendungszweck hin entwickelt worden sind. Hierzu gehören beispielsweise die Abbildung von Kennlinien durch neuronale Netze oder die Abbildung von Fehlermodellen durch Gauß-Prozesse.

Wie oben beschrieben ist das Ziel des maschinellen Lernens aus den vorhandenen Daten brauchbare Informationen zu extrahieren um anschließend Vorhersagen über unbekannte Eigenschaften und Vorschläge für Entscheidungen treffen zu können (siehe auch [5]). Dabei ist es erst einmal nicht von Bedeutung, was die betrachteten Daten genau beschreiben. ML besteht grundsätzlich aus drei verschiedenen Komponenten, die in der Lage sind das ”Lernen” abzubilden. Dazu gehören

- ein Datensatz,
- ein mathematisches Modell und
- ein Lernalgorithmus.

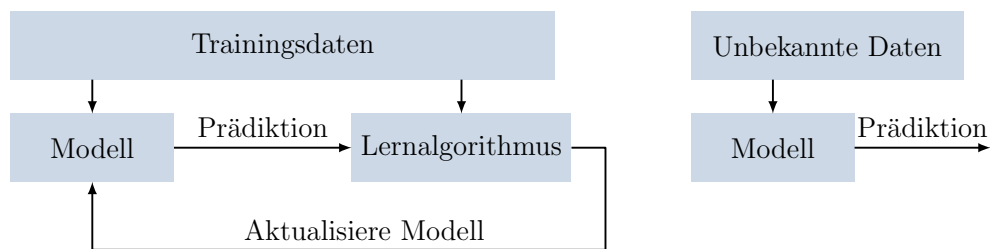
Der Datensatz beinhaltet dabei die Informationen, die das betrachtete System beschreiben und muss so bestimmt werden, dass die Lernaufgabe durchführbar ist. Informationen, die nicht in den Daten vorhanden sind, können durch das Modell abgebildet werden. Das mathematische Modell soll in einer geeigneten Weise die gelernte Information speichern können und nach dem Lernen für Vorhersagen genutzt werden können. Ein solches Modell kann beispielsweise ein neuronales Netz oder ein Entscheidungsbaum sein. Der Lernalgorithmus trainiert das Modell mit Hilfe der Daten, um die Aussagen der Daten verallgemeinern zu können und es dem Modell zu ermöglichen nach dem Lernen auch Prädiktionen für unbekannte Daten durchführen zu können. Es ist darauf zu achten, dass diese drei Komponenten aufeinander und auf die spezifische Lernaufgabe abgestimmt werden.



**Abbildung 2.3** Einordnung verschiedener Kategorien des maschinellen Lernens.

Methoden des maschinellen Lernens lassen sich nach ihren Eigenschaften in unterschiedliche Kategorien einordnen, siehe auch Abbildung 2.3. Eine Möglichkeit der Kategorisierung ist die Art von Daten, die zum Lernen zur Verfügung stehen. So macht es einen Unterschied, ob ein Algorithmus entwickelt wird, der anhand von vorgegebenen Ein- **und** Ausgangsdaten weiß, was er abbilden soll oder ob keine gewünschten Ausgangsdaten vorhanden sind. Diese unterschiedlichen Szenarien werden als *überwachtes* bzw. *unüberwachtes* Lernen bezeichnet. Methoden des unüberwachten Lernens versuchen dabei nur anhand der Eingabedaten Muster zu erkennen und diesen beispielsweise Kategorien zuzuordnen, so dass hier Methoden vorliegen, die keine Zielwerte der Daten benötigen. Ein Beispiel dieser Art von Methoden ist die Clusteranalyse, bei der es das Ziel ist, Cluster in unstrukturierten Daten zu finden, die auf Gemeinsamkeiten in den Daten beruhen. Da für die Arbeiten der Nachwuchsgruppe DART Algorithmen des überwachten Lernens die größte Bedeutung haben, soll an dieser Stelle aber hauptsächlich auf diese Art des Lernens eingegangen werden. Für das überwachte Lernen müssen Daten mit sogenannten Labels vorliegen. Das heißt, dass diese Datensätze vorab durch z.B. Expertenwissen

oder Experimente um die gewünschte Ausgabe, die Label, für eine definierte Eingabe ergänzt werden, sodass auch die korrekte Ausgabe eines Modells überprüft werden kann. Dies ist in Abbildung 2.4 auf der linken Seite abgebildet. Der Lernalgorithmus erhält die Prädiktion der Ausgabe aus dem bisher gelernten Modell und kann sie mit der Sollausgabe aus den Trainingsdaten vergleichen. Wenn die Werte gut genug übereinstimmen, kann das Lernen beendet werden. Im anderen Fall macht der Lernalgorithmus einen Änderungsvorschlag für das Modell, welches dementsprechend aktualisiert wird. Damit ist dann eine Lerniteration abgeschlossen. Nach dem Training des Modells kann es im Idealfall für die Prädiktion von unbekannten Daten, bei denen die Ausgabe nicht bekannt ist, genutzt werden. Dies ist in Abbildung 2.4 rechts dargestellt. Wenn dies gut gelingt, spricht man von einem Modell, das generalisiert, da es für unbekannte Eingabedaten korrekte Ausgaben erzeugt. Das heißt, eine hohe *Generalisierbarkeit* bedeutet, dass das Modell die Fähigkeit besitzt angemessen mit neuen bisher unbekannten Daten umzugehen, die aus demselben System stammen wie die, die zur Erstellung des Systems verwendet wurden.



**Abbildung 2.4** Darstellung des überwachten Lernprozesses: Training des Modells (links); Verwendung des Modells für die Prädiktion bei einem unbekannten Datensatz (rechts), angelehnt an [5].

Innerhalb des Gebiets des überwachten Lernens gibt es weitere Unterscheidungen der Methoden, abhängig davon wie das Lernergebnis des Modells aussehen soll, siehe auch Abbildung 2.3. Für die Nachwuchsgruppe DART sind dabei Modelle, die Regressionsaufgaben lösen, die wichtigste Klasse. Eine Regression liegt dann vor, wenn das Modell nach dem Lernprozess einen funktionalen Zusammenhang zwischen Ein- und Ausgangsgrößen beschreibt. Oder auch in einer mehr mathematischen Beschreibung: Die Regression ist ein statistisches Analyseverfahren, welches Beziehungen zwischen einer abhängigen und einer oder mehrerer unabhängiger Variablen in Form von  $\mathbf{y} = f(\mathbf{x})$  modelliert [6]. Dies kann beispielsweise eine Kennlinie sein, bei der den Eingangsdaten die gewünschten Ausgangsgrößen in einem passenden mathematischen Zusammenhang zugeordnet werden oder ein neuronales Netz, das für eine Funktionsapproximation genutzt wird. Eine weitere Kategorie im überwachten Lernen ist die Klassifikation. Hier wird ein Modell mittels maschinellen Lernens so gelernt, dass es Daten in eine planmäßige Sammlung von abstrakten Klassen, die zur Abgrenzung und Ordnung verwendet werden, sortieren kann [2]. Dabei werden die einzelnen Klassen durch die Einteilung von Objekten anhand bestimmter übereinstimmender Merkmale bestimmt. Zentraler Punkt hierbei ist das Erkennen von Mustern, den Merkmalen, die allen Dingen einer Kategorie gemeinsam sind und sie vom Inhalt anderer Kategorien unterscheiden. Für eine Klassifikation gibt es viele Beispiele, die auch schon aus dem Alltag bekannt sind. Dazu

gehören Spracherkennung, Texterkennung oder auch Gesichtserkennung, also Aufgaben, die der Mensch standardmäßig erfüllen kann. Im technischen Bereich kann eine Klassifizierung auch die Fälle "normaler" bzw. "fehlerhafter" Betrieb eines Systems beschreiben und damit sehr hilfreich in der Fehlererkennung sein. Eine weitere Kategorie des überwachten Lernens ist das Deep Learning (siehe Abbildung 2.3), was auf der Verwendung von tiefen neuronalen Netzen beruht [7]. Die beim Deep Learning verwendeten neuronalen Netze besitzen eine große Anzahl von versteckten Schichten zwischen Eingabe- und Ausgabeschicht und damit eine komplexe innere Struktur, benötigen aber für das Training dementsprechend eine große Menge an Daten. Durch ihre sehr hohe Komplexität erlaubt Deep Learning die Verarbeitung und Analyse von sehr komplexen Datenmustern. Aktuell werden große Modelle wie *Large Language Models* betrachtet, die bis zu einer Billion lernbarer Parameter besitzen und es ermöglichen sollen eine generelle künstliche Intelligenz wie beispielsweise ChatGPT [8] zu erzeugen. Viele Wissenschaftler\*innen sehen hierin einen großen technologischen Fortschritt, der auch gesellschaftliche Auswirkungen haben wird.

Eine weitere Form des maschinellen Lernens, welche zwischen dem überwachten und unüberwachten Ansatz steht, ist das Reinforcement Learning<sup>1</sup> (RL) oder auch bestärkendes Lernen, siehe auch Abbildung 2.3. Die Grundidee des RL basiert darauf, dass ein System sein Verhalten dahingehend anpasst, dass es eine möglichst große Belohnung in der Interaktion mit seiner Umwelt erhält, siehe hierzu auch [9]. Das heißt, wir gehen davon aus, dass ein Agent erst einmal frei in einer vorgegebenen Umwelt agieren kann und unterschiedliche Aktionen durchführen kann. Durch die Interaktion erhält er Informationen darüber, welche Konsequenzen seine Handlungen haben und lernt durch Ausprobieren, was für sein Ziel vorteilhafte oder unvorteilhafte Handlungen sind. Dieses Lernen von Interaktionen ist ein sehr natürliches Verständnis von diesem Vorgang, denn es beschreibt auch die Lernansätze, die der Mensch verfolgt, nämlich Lernen durch Ausprobieren und Erkunden. Die schwierigsten Aufgaben beim RL liegen dann vor, wenn eine Aktion nicht direkt eine gute Bewertung erhält, sondern sich erst in der noch unbekannten Zukunft positiv auswirkt. Das finale Ziel der RL Algorithmen wird aber dadurch beschrieben, dass der Agent die maximale Gesamtbelohnung erzielen soll und daher sind temporäre schlechte Handlungen möglicherweise in der Zukunft trotzdem vorteilhaft. Dies führt zu einem weiteren wichtigen Problem, welches im RL gelöst werden muss, und zwar einen Kompromiss zwischen Exploitation (Ausnutzen) und Exploration (Erkunden) zu finden. Im ersten Fall nutzt man die Aktionen, in denen gute Ergebnisse schon bekannt sind, um eine möglichst hohe Belohnung zu erhalten und im anderen Fall erkundet man unbekannte Bereiche, in denen möglicherweise noch bessere Ergebnisse zu erzielen sind. Hier wird deutlich, dass dies ein Dilemma ist, welches durch den jeweiligen Algorithmus adressiert werden muss.

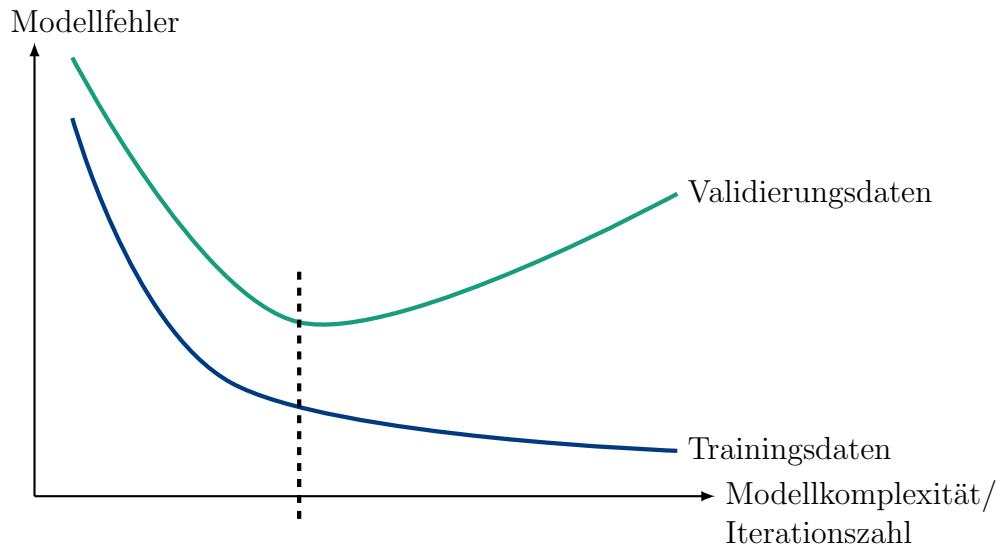
Für das maschinelle Lernen gibt es Empfehlungen, wie der Gesamtdatensatz am effektivsten für ein möglichst gutes Lernergebnis zu nutzen ist. Mit circa 70 % des Gesamtdatensatzes sollten die Trainingsdaten den größten Teil beinhalten. Die Trainingsdaten werden direkt für das Lernen des gesuchten Zusammenhangs verwendet und sollten nach Möglichkeit alle Effekte und Einflüsse der Lernaufgabe beinhalten. Die Testdaten sollten circa 15 % des Gesamtdatensatzes beinhalten und dienen dem

---

<sup>1</sup>Da der englischsprachige Begriff allgemein üblich für diese Methoden ist, wird er in dieser Arbeit bevorzugt verwendet.

abschließenden Test des aufgestellten Modells. Das Testen besteht also daraus, mit Hilfe von unbekannten Daten die Güte der Vorhersage des Modells überprüfen zu können. Mit den übrigen 15 % wird das Modell während des Lernprozesses validiert. Abhängig von den Ergebnissen mit diesem Datensatz werden also die Architektur oder andere Hyperparameter des Modells angepasst. Ein Beispiel hierfür bei neuronalen Netzen ist die Anpassung der Anzahl von Neuronen oder versteckten Schichten. Damit stehen die Validierungsdaten zwischen Trainings- und Testdaten, haben aber ebenfalls einen großen Einfluss auf das Lernergebnis. Bevor Messdaten jedoch als Trainings-, Validierungs- oder Testdaten genutzt werden können, ist u. U. eine *Vorverarbeitung der Daten* erforderlich, um die in den Daten enthaltenen Informationen nutzen zu können [10], [11]. So erfolgt in der Regel eine Bereinigung der Rohdaten, indem z. B. Rauschen gefiltert, Ausreißer detektiert und Duplikate erkannt werden. Liegen unterschiedlich skalierte Datensätze vor, müssen diese normalisiert oder standardisiert werden, um sie verarbeiten zu können. Ebenso kann es notwendig sein, dass die Rohdaten in ein geeignetes Format übertragen oder aufgrund verschiedener Datenquellen zu einem Datensatz fusioniert werden müssen. Ferner kann es sinnvoll sein, bereits in der Datenvorverarbeitung eine Merkmalsanalyse vorzunehmen, um ggf. eine Reduktion auf die wesentlichen Daten zu erhalten. Die Vorverarbeitung der Daten ist daher ein wichtiges Werkzeug, um eine hohe Datenqualität und die erfolgreiche Nutzung nachfolgender Anwendungen zu ermöglichen.

Schon für den Begriff der Generalisierbarkeit wurde beschrieben, dass ein zentrales Ziel beim maschinellen Lernen die Güte der Vorhersagefähigkeit des gelernten Modells auf unbekannten Daten ist. Dabei ist es aber nicht sinnvoll ein Lernmodell mit einer möglichst hohen Komplexität auszuwählen, bei dem sehr viele Hyperparameter bestimmt werden müssen. Denn bei einer zu hohen Modellkomplexität im Verhältnis zu den vorhandenen Daten besteht die Gefahr, dass sich das Modell zu stark auf die Trainingsdaten anpasst und auch ungewünschte Eigenschaften wie zufällige Besonderheiten und Fehler in den Daten gelernt werden. Dieses Verhalten widerspricht einer guten Generalisierung und wird mit *Overfitting* (oder auch Überanpassung) bezeichnet. Erkennbar ist Overfitting daran, dass das Lernen sehr gute Ergebnisse auf den Trainingsdaten zeigt, aber ein Test auf den Validierungsdaten schlecht ausfällt, siehe Abbildung 2.5 rechts der gestrichelten Linie. Die Empfehlung ist dann die Modellkomplexität so zu wählen, dass der Fehler auf den Validierungsdaten minimal ist. Ein ähnlicher Effekt des Overfitting tritt auf, wenn zu viele Trainingsiterationen durchgeführt werden und auch in diesem Fall sollte das Training im Minimum des Fehlers der Validierungsdaten abgebrochen werden. Weiteres Training führt dazu, dass ungewünschte Effekte in den Trainingsdaten gelernt werden, dass Modell aber nicht besser in seiner Generalisierbarkeit wird. Overfitting kann durch eine Kreuzvalidierung vermieden werden. Bei dieser Methode wird das Training des Modells mehrfach auf einem unterschiedlich aufgeteilten Datensatz durchgeführt, so dass am Ende verschiedene Modelle verglichen werden können. Das heißt, es werden  $n$  verschiedene Datensätze aus dem Gesamtdatensatz erzeugt und diese werden immer wieder unterschiedlich in die drei Klassen Training, Validierung und Test aufgeteilt. Final werden die Modelle miteinander verglichen und wenn die Modelle vergleichbar sind, dann kann angenommen werden, dass der verwendete Algorithmus für den vorliegenden Datensatz generalisierte Modell erzeugt. [12]



**Abbildung 2.5** Typischer Effekt des Overfittings: Der Fehler auf den Trainingsdaten sinkt bei höherer Modellkomplexität bzw. weiteren Trainingsiterationen immer weiter. Das Ergebnis auf den Validierungsdaten wird aber ab der gestrichelten Linie wieder schlechter. Hier tritt Overfitting auf.

Zur Vermeidung von **Overfitting** sollten die Ergebnisse eines Modells auf Trainings-, Validierungs- und Testdaten betrachtet werden und auf Overfitting kontrolliert werden. Sind diese vergleichbar und für den benötigten Anwendungsfall ausreichend gut, kann eine belastbare Entscheidung über den Einsatz des Modells getroffen werden. [12]

Die Modelle des ML besitzen im Allgemeinen Parameter, die ihre jeweilige Ausprägung beschreiben. Dies kann zum Beispiel die Anzahl der Neuronen und versteckten Schichten in einem neuronalen Netz oder auch die Wahl der Aktivierungsfunktionen sein. Diese Parameter werden *Hyperparameter* genannt und da sie großen Einfluss auf das Lernergebnis haben, müssen sie sehr sorgfältig ausgewählt werden. So ist es das Ziel einer *Hyperparameteroptimierung* die besten Einstellungen für diese vorab festgelegten Parameter des Modells zu finden. Eine Einstellung durch einen Menschen ist dabei möglich, aber mit steigender Anzahl der Hyperparameter wird die Dimension des Parameterraums sehr hoch und es ist nicht mehr möglich die Einflüsse einzelner Parameter korrekt abzuschätzen. Es wird daher empfohlen, die Hyperparameteroptimierung strategisch durch einen Algorithmus durchzuführen, zum Beispiel durch Ansätze, die im einfachen Fall auf einer Zufallssuche oder auf sequentiellen Ansätzen mit Nutzung des Expected Improvement basieren. In Forschungsarbeiten konnte gezeigt werden, dass es sehr effektiv ist, in bestehenden Methoden die Konfigurationen zu optimieren und dadurch verbesserte Ergebnisse zu erhalten. Daher empfehlen [13], dass die Hyperparameteroptimierung immer als eine formale äußere Schleife im Lernprozess durchgeführt werden sollte.

Um Phänomene wie das Over- oder Underfitting zu beeinflussen, die Anzahl der Hyperparameter zu reduzieren oder Eigenschaften dieser zu forcieren, werden häufig Strategien der *Regularisierung* oder des *Feature Engineering* genutzt. Grundlegen-

des, gemeinsames Ziel all jener Techniken ist es, dass das resultierende Modell einem konkreten Zweck dienen kann, z. B. dass es gut generalisiert, indem Over- und Underfitting vermieden werden, oder dass es interpretierbar und übersichtlich bleibt, indem wenige Hyperparameter genutzt werden. Eine bekannte Regularisierungsstrategie, um Overfitting in neuronalen Netzen zu vermeiden, stellt beispielsweise das Dropout dar, welches während des Trainings (häufig zufallsbasiert) eine bestimmte Menge an Neuronen an- und abschaltet [10]. Noch populärer ist die  $\ell_1$ -Regularisierung, welche in vielfältigen Methoden angewendet wird. Durch den Einsatz eines  $\ell_1$ -Terms in Kombination mit einer quadratischen, meist  $\ell_2$ -basierten Gütefunktion wird die Nutzung weniger Hyperparameter und dadurch die Verringerung der Modellkomplexität gefördert [14]. Ein prominentes Beispiel stellt die LASSO-Regularisierung dar, welche für das SINDy-Verfahren verwendet wird (vgl. Abschnitt 3.2). Verfahren des Feature Engineering zielen zumeist auf eine Modellordnungsreduktion ab, um die Komplexität des Modells zu reduzieren und es interpretierbarer zu gestalten. Dazu wird in der Regel eine Merkmalsanalyse durchgeführt, um Charakteristika zu identifizieren, die die wesentlichen Eigenschaften eines dynamischen Systems beschreiben und folglich im Modellverhalten beibehalten werden, wohingegen weitere vernachlässigt oder zusammengefasst werden können. Eine der populärsten Methoden ist die Hauptkomponentenanalyse [15], im Englischen Principal Component Analysis (PCA), welche basierend auf einer Eigenwertanalyse relevante Einsichten zur Verfügung stellt (vgl. Unterabschnitt 5.3.2). Nachdem die grundlegenden Prinzipien datenbasierter Methoden vorgestellt worden sind, wird im folgenden Abschnitt genauer beschrieben, wie diese Methoden zielgerichtet im hybriden Regelungsentwurf eingesetzt werden sollen.

### 2.3 Zielsetzung beim hybriden Regelungsentwurf

In Abschnitt 1.3 wurde hervorgehoben, welche Vorteile die Nutzung datengetriebener Methoden in der Regelungstechnik erzielen können. Da bereits im klassischen Regelungsentwurf Messdaten verwendet werden, um physikalisch motivierte Modelle zu parametrieren oder Systeme zu identifizieren (vgl. Abbildung 1.3), ist es sinnvoll, diese Daten auch in größerem Umfang zur hybriden Modellierung zu nutzen. Übergeordnetes Ziel dieses Vorgehens ist es, Modelle und Regelungsansätze zu verbessern sowie adaptiv gestalten zu können, aber gleichzeitig physikalische Einsicht in das System zu konservieren oder vertiefend zu erhalten. Dies bedeutet insbesondere, dass die Entwicklung reiner Black-Box-Modelle vermieden werden soll. Weil jeder klassische Regelungsentwurf zunächst mit dem Schritt der Modellbildung beginnt (vgl. Abbildungen 1.2 und 2.6), steht im hybriden Regelungsentwurf die Entwicklung eines hybriden Modells im Fokus. Dieses soll eine *regelungstechnisch verwertbare Form* aufweisen (vgl. Dissertation [16]), welche anhand der folgenden Kriterien charakterisiert wird (vgl. Dissertationen [16], [17], [18]).

**Geringer Modellierungsaufwand.** Aufgrund verkürzter Entwicklungszyklen ist es wichtig, dass die Modellbildung möglichst ressourceneffizient erfolgt. Ein zu detailliertes Modell kann im schlechtesten Fall sogar den zeitlichen Vorteil, der durch den modellbasierten Entwurf erreicht wird, durch den erhöhten Aufwand



kompensieren [19], [20]. Daher soll der Modellierungsaufwand nur so hoch wie nötig sein.

**Geringer Rechenaufwand bei der Prädiktion.** Modellprädiktive Regelungsansätze optimieren anhand des Streckenmodells den Ausgangsgrößenverlauf mittels des Stellgrößenverlaufs. Ebenso nutzen modellbasierte Zustandsbeobachter das Streckenmodell, um die Zustände des Systems zu präzisieren und basierend auf Messungen zu korrigieren. Dies sind nur zwei Beispiele, bei denen Echtzeitfähigkeit eine wichtige Rolle spielt. Aus diesem Grund soll das hybride Modell möglichst kostengünstig präzisiert werden können [21].

**Hohe Dateneffizienz.** Das Sammeln von Daten ist an mechatronischen Prüfständen aufgrund des hohen Zeitaufwands, des Energiebedarfs und des Ressourcenverbrauchs aufwendig. Daher ist es erstrebenswert, dass die datengetriebenen Modellbildungsansätze auch bei begrenzten Datenmengen eine effiziente Generalisierungsfähigkeit zeigen.

**Physikalische Interpretierbarkeit.** Ein physikalisch interpretierbares Modell ermöglicht ein tiefes Verständnis der zugrunde liegenden Prozesse im gesteuerten System. Dadurch wird eine zuverlässige Vorhersage des Systemverhaltens erleichtert, was insbesondere in sicherheitskritischen Anwendungen entscheidend ist. Folglich soll das datengetriebene Modell eine Form aufweisen, die sich physikalisch analysieren lässt, beispielsweise durch Stabilitätsuntersuchungen.

**Nutzung von physikalischem Vorwissen.** Zudem wird angenommen, dass die Person, die das regelungstechnische System entwickelt, über Vorwissen bezüglich grundlegender physikalischer Zusammenhänge von mechatronischen Systemen verfügt. Im Kontext der Zielsetzung dieses Forschungsvorhabens soll dieses physikalisch motivierte Vorwissen in die datengetriebene Modellbildung integriert werden<sup>2</sup>, um eine optimale Nutzung der Vorteile und Stärken beider Ansätze zu erreichen [22].

**Anwendung bestehender Entwurfsverfahren.** Bestehende Verfahren für den Regler- und Beobachterentwurf zeichnen sich durch ihre systematische Vorgehensweise aus, die auf physikalischen Prinzipien und systemtheoretischen Anforderungen wie Stabilität oder Steuerbarkeit basiert [21], [24]. Im Gegensatz dazu erweisen sich rein datengetriebene Streckenmodelle aufgrund der Black-Box-Struktur als nicht unmittelbar für den Regler- und Beobachterentwurf nutzbar. Dies führt zu Herausforderungen bei der Erreichung bestimmter Regelungsziele. Demnach ist es erstrebenswert, dass das Modell eine geradlinige Anwendung bestehender Entwurfsverfahren ermöglicht.

**Adaptionsfähigkeit bei Systemveränderungen.** Während des Produktlebenszyklus eines mechatronischen Systems treten vorhersehbare oder unvorhersehbare Veränderungen der Dynamik auf, z. B. durch den Einbau neuer Bauteile,

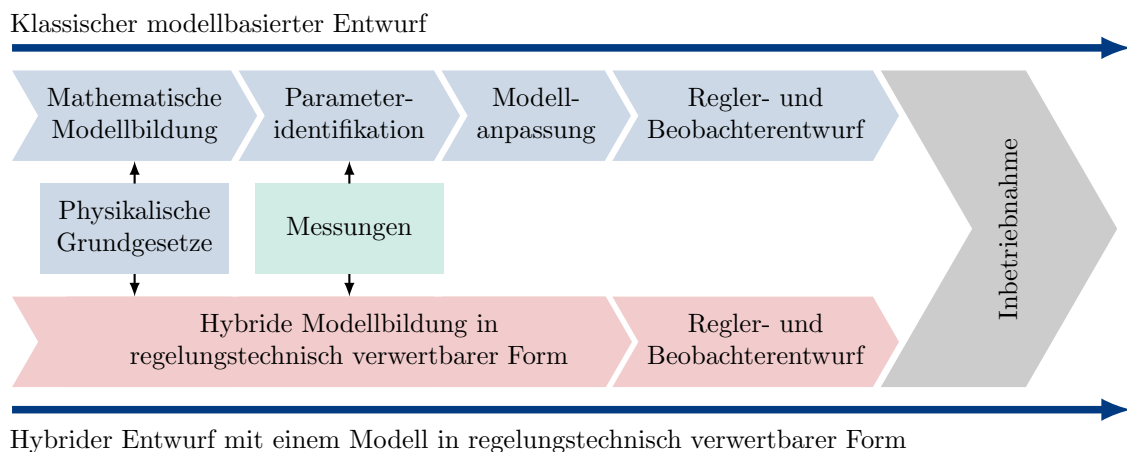
---

<sup>2</sup>Die Berücksichtigung von physikalischen Grundgesetzen beim maschinellen Lernen wird auch als *Physics-informed machine learning* bezeichnet und erfährt im wissenschaftlichen Kontext eine weitreichende Beachtung, vgl. [22], [23].

Verschleiß oder Temperaturschwankungen. Zusätzlich unterliegt das verwendete Modell in der Regel Unsicherheiten, beispielsweise durch fehlerhafte Annahmen. Weil durch die Rückkopplungsstruktur regelungstechnischer Systeme während des Betriebs ohnehin laufend Messdaten erfasst werden, ist es reizvoll, das Streckenmodell online anzupassen. Daher sollten die datengetriebenen Modelle adaptiv sein.

**Gutes Extrapolationsverhalten.** Ein Modell mit gutem Extrapolationsverhalten erlaubt zuverlässige Prädiktionen auch in unbekannten, neuen Gebieten des Phasenraums. Aufgrund der Rückführungsstruktur regelungstechnischer Systeme ist es daher erforderlich, dass ein hybrides Modell in der Lage ist, neue, bisher ungesehene Datensätze außerhalb der Trainingsmenge zu verarbeiten, und folglich eine gute Generalisierbarkeit aufweist [10], [11].

Diese Ziele stellen die wesentlichen Anforderungen an ein hybrides Modell dar, welches für regelungstechnische Anwendungen genutzt werden soll. Die Abbildung 2.6 veranschaulicht das angestrebte hybride Vorgehen als Alternative zum klassischen modellbasierten Entwurf, dessen aufwendiger, u. U. iterativer Prozess vereinfacht werden soll. So erlaubt die hybride Modellbildung eine Modellierung in regelungstechnisch wertvoller Form, welche ebenfalls basierend auf physikalischen Gesetzen und Messdaten gebildet wird, jedoch ohne mehrteilige Modellierungsschritte auskommt und direkt für den Regler- und Beobachterentwurf zugänglich ist. Daher werden im folgenden Kapitel einige Verfahren zur hybriden Modellierung vorgestellt, die den zuvor genannten Anforderungen entsprechen.



**Abbildung 2.6** Der hybride Entwurf mit einem Modell in regelungstechnisch verwertbarer Form dient als Alternative zum klassischen modellbasierten Regelungsentwurf [16], [25].

# 3 Modellbildung

Um ein technisches System zu steuern oder zu regeln, ist grundlegendes Verständnis über dieses erforderlich. In der Regelungstechnik wird daher zunächst ein Modell der Regelstrecke<sup>1</sup> angefertigt, welches die wesentlichen Eigenschaften dieser abbilden soll. Klassischerweise werden physikalische Grundgesetze verwendet, um Differentialgleichungen oder Übertragungsfunktionen herzuleiten. Gleichzeitig dienen Messdaten als Ergänzung, um beispielsweise Parameter dieser physikalischen Modelle zu identifizieren. Neben diesem analytischen Vorgehen können aber auch ausschließlich Messdaten verwendet werden, um beispielsweise experimentell eine Übertragungsfunktion durch die Analyse der Systemantwort zu bestimmen [1], [21]. In Abschnitt 1.3 sind diese beiden Perspektiven der Modellbildung bereits thematisiert worden. Sowohl die analytische als auch die experimentelle Modellierung finden jeweils berechnete Anwendung in der regelungstechnischen Modellbildung und weisen verschiedene Stärken auf (vgl. Abbildung 1.3), sodass eine Kombination zu hybriden Modellen besonders vorteilhaft ist. Vor allem in der Situation von Modellungenauigkeiten, welche wegen getroffener Annahmen und Vereinfachungen existieren und von Parameterunsicherheiten bis zu fehlenden Modelltermen reichen können, und Systemveränderungen, die z. B. in Abhängigkeit von Verschleiß oder Temperaturschwankungen auftreten können, leisten hybride Modelle ihren Beitrag zur präzisen Systembeschreibung. Dabei kennzeichnen der Grad des verwendeten physikalischen Vorwissens und das Maß an genutzten Messdaten die Ausprägung sogenannter White-, Gray- oder Black-Box-Modelle (vgl. Abbildung 1.4), wobei aufgrund der erforderlichen systemtheoretischen Analyse und Verwendung in bestehenden regelungstechnischen Entwurfsverfahren keine Black-Box-Modelle im Rahmen der Nachwuchsgruppe DART angestrebt wurden (vgl. Abschnitt 2.3).

In diesem Kapitel werden folglich hybride Modellierungstechniken thematisiert, welche sowohl physikalisch motivierte als auch datengetriebene Komponenten verwenden. Dabei werden einerseits Gesamtmodelle vorgestellt, die Komponenten beider Perspektiven aufweisen, und andererseits hybride Fehlermodelle diskutiert, die in serieller Komplettierung ein datengetriebenes Restmodell zu einem physikalischen Modell ergänzen. Ausgangspunkt aller dieser hybriden Methoden ist die Berücksichtigung von Vorwissen, welches zumindest die Kenntnis der Systemordnung, der Zustands-, Ein- und Ausgangsgrößen umfasst, manchmal aber auch ein einfaches physikalisches Modell beinhalten kann, und die Verfügbarkeit von Messdaten, die in Form einer Zeitreihe vorliegen und in Matrixform gebracht werden können. Ziel dieser Methoden ist es, eine hohe Modellgüte zu erzielen und gleichzeitig eine Form zu konservieren, die eine systemtheoretische Analyse bzw. physikalische Interpretierbarkeit erlaubt. Neben einigen grundlegenden datengetriebenen Verfahren in den Abschnitten 3.2 und 3.3.1 werden insbesondere die in der Nachwuchsgruppe DART er-

---

<sup>1</sup>Das zu regelnde oder zu steuernde System wird als (Regel-)Strecke bezeichnet.

arbeiteten hybriden Methoden vorgestellt. Diese umfassen Koopman-basierte Techniken, physikalische motivierte neuronale Netze, datengetriebene strukturerhaltende Verfahren sowie probabilistische hybride Modellierungsstrategien und werden durch einfache Anwendungsbeispiele illustriert. Darüber hinaus werden Vor- und Nachteile jener Methoden aufgezeigt und bzgl. ihrer Modellgüte und der in Abschnitt 2.3 formulierten Anforderungen analysiert.

## 3.1 Methoden basierend auf dem Koopman-Operator

B. O. Koopman formulierte 1931 eine operatortheoretische Perspektive zur linearen Beschreibung der Dynamik nichtlinearer Systeme durch die Beobachtung in einem im Allgemeinen höherdimensionalen Raum [26]. Diese lineare Systembeschreibung führt zu einer erheblichen Reduktion des Rechenaufwands bei der Prädiktion des Systemverhaltens und bietet zudem den Vorteil, dass sich bewährte Entwurfsverfahren aus der linearen Regelungstechnik, z. B. Riccati-Regler, direkt auf nichtlineare Systeme anwenden lassen. Im folgenden Abschnitt wird die Grundidee des Koopman-Operators in Anlehnung an [27], [28] eingeführt. Anschließend werden Ansätze zur Modellbildung basierend auf dem Koopman-Operator erläutert und anhand von Beispielen illustriert.

### 3.1.1 Theoretische Einordnung des Koopman-Operators

Im Folgenden werden zunächst zeitkontinuierliche autonome dynamische Systeme

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}) \quad (3.1)$$

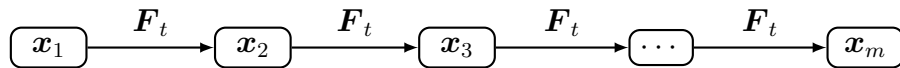
betrachtet, wobei  $\mathbf{x} \in \mathbb{R}^n$  ein  $n$ -dimensionaler Zustand und  $\mathbf{f} : \mathbb{R}^n \rightarrow \mathbb{R}^n$  eine Lipschitz-stetige Funktion sind. Für eine Zeitspanne  $t$  lässt sich der Fluss  $\mathbf{F}_t : \mathbb{R}^n \rightarrow \mathbb{R}^n$  definieren, der den Zustand  $\mathbf{x}(t_0)$  um einen Zeitschritt vorwärts auf den Zustand  $\mathbf{x}(t_0 + t)$  abbildet

$$\mathbf{F}_t(\mathbf{x}(t_0)) = \mathbf{x}(t_0 + t) = \mathbf{x}(t_0) + \int_{t_0}^{t_0+t} \mathbf{f}(\mathbf{x}(\tau)) \, d\tau, \quad (3.2)$$

sodass sich mit  $k \in \mathbb{Z}$  das zeitdiskrete dynamische System

$$\mathbf{x}_{k+1} = \mathbf{F}_t(\mathbf{x}_k) \quad \text{mit} \quad \mathbf{x}_k = \mathbf{x}(k\Delta t) \quad (3.3)$$

ergibt, vgl. Abbildung 3.1.



**Abbildung 3.1** Der Fluss  $\mathbf{F}_t$  beschreibt die zeitliche Entwicklung des Zustands  $\mathbf{x}$  von einem Zeitschritt in den darauffolgenden Zeitschritt.

Für eine beliebige Observable<sup>2</sup>  $g : \mathbb{R}^n \rightarrow \mathbb{R}$  ist der lineare Koopman-Operator  $\mathcal{K}_t$

<sup>2</sup>Der Begriff der Observable lässt sich hier als Beobachtung bzw. Messgröße des Systems verstehen.

definiert als Komposition mit dem Fluss

$$\mathcal{K}_t g := g \circ \mathbf{F}_t. \quad (3.4)$$

Damit ergibt sich für das zeitdiskrete System (3.3)

$$\mathcal{K}_t g(\mathbf{x}_k) = g(\mathbf{F}_t(\mathbf{x}_k)) = g(\mathbf{x}_{k+1}), \quad (3.5)$$

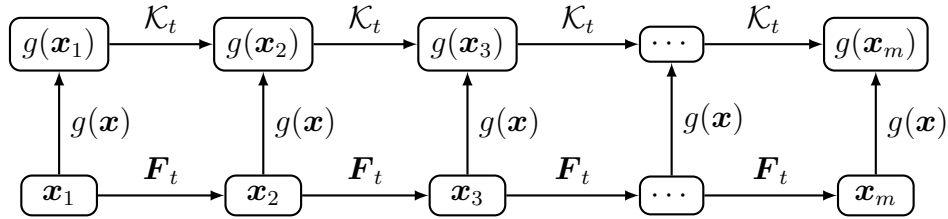
d. h. der Koopman-Operator überführt die Observable  $g(\mathbf{x}_k)$  in den nächsten Zeitschritt  $g(\mathbf{x}_{k+1})$ , vgl. Abbildung 3.2. Der Koopman-Operator  $\mathcal{K}_t$  ist ein linearer unendlichdimensionaler Operator. Für hinreichend glatte Systeme lässt sich ein zeitkontinuierliches Analogon

$$\mathcal{K}g = \frac{d}{dt}g, \quad (3.6)$$

definieren, wobei  $\mathcal{K}$  der infinitesimale Generator mit

$$\mathcal{K}g = \lim_{t \rightarrow 0} \frac{\mathcal{K}_t g - g}{t} = \lim_{t \rightarrow 0} \frac{g \circ \mathbf{F}_t - g}{t} \quad (3.7)$$

ist [28].



**Abbildung 3.2** Veranschaulichung des Koopman-Operators.

Anhand eines Beispielsystems [27] lässt sich die Idee des Koopman-Operators veranschaulichen. Das betrachtete zeitkontinuierliche nichtlineare dynamische System wird durch die Differentialgleichungen

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} \mu x_1, \\ \lambda (x_2 - x_1^2) \end{bmatrix} \quad \text{mit} \quad \lambda, \mu \in \mathbb{R}, \lambda \ll \mu < 0 \quad (3.8)$$

beschrieben. Es handelt sich um eine Dynamik mit einer langsamen Mannigfaltigkeit  $x_2 = x_1^2$ , weil der Zustand  $x_2$  sich zeitlich schneller entwickelt als der Zustand  $x_1$ . Um eine lineare Systemdarstellung mittels des Koopman-Operators zu erhalten, werden die Observablen geschickt so festgelegt, dass sie den originalen Zustandsvektor  $\mathbf{x} = [x_1, x_2]^\top$  und zusätzlich die Nichtlinearität  $x_1^2$  enthalten<sup>3</sup>

$$\mathbf{g}(\mathbf{x}) = \begin{bmatrix} g_1(\mathbf{x}) \\ g_2(\mathbf{x}) \\ g_3(\mathbf{x}) \end{bmatrix} = \begin{bmatrix} x_1 \\ x_2 \\ x_1^2 \end{bmatrix}. \quad (3.9)$$

Bezüglich der Observablen  $\mathbf{g}(\mathbf{x})$  lässt sich die Systemdynamik durch die Berechnung

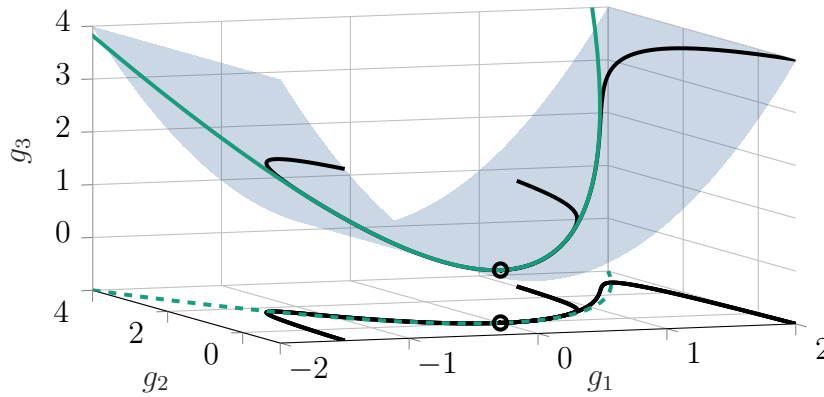
<sup>3</sup>Dieser Ansatz ist ähnlich zur *Carleman-Linearisierung*, vgl. [29], [30].

### 3 Modellbildung

der zeitlichen Ableitungen unter Berücksichtigung der Kettenregel beschreiben durch

$$\underbrace{\begin{bmatrix} \dot{g}_1 \\ \dot{g}_2 \\ \dot{g}_3 \end{bmatrix}}_{\dot{g}} = \underbrace{\begin{bmatrix} \mu & 0 & 0 \\ 0 & \lambda & -\lambda \\ 0 & 0 & 2\mu \end{bmatrix}}_{\mathcal{K}} \underbrace{\begin{bmatrix} g_1 \\ g_2 \\ g_3 \end{bmatrix}}_g. \quad (3.10)$$

Die Abbildung 3.3 veranschaulicht die Dynamik des eigentlich zweidimensionalen nichtlinearen Originalsystems im dreidimensionalen Raum der Observablen.



**Abbildung 3.3** Visualisierung von Beispieltrajektorien für  $\mu = -0,05, \lambda = -1$ . Die parabelförmige Fläche ist die Zwangsbedingung  $g_3(x) = x_1^2$ , auf der alle Trajektorien verlaufen. In grün ist die attraktive langsame Mannigfaltigkeit des Systems dargestellt, der sich alle Trajektorien zunächst schnell annähern, bevor sie danach langsam in den Ursprung streben.

Das beschriebene Vorgehen ist bei den meisten auftretenden nichtlinearen dynamischen Systemen nicht ohne Weiteres möglich. Dieser Zusammenhang lässt sich anhand des Beispielsystems

$$\dot{x} = x^2 \quad (3.11)$$

verdeutlichen, bei dem die alleinige Erweiterung des Originalzustandes um die Nichtlinearität zu einer unendlichdimensionalen Systembeschreibung führen würde

$$\begin{bmatrix} \dot{g}_1 \\ \dot{g}_2 \\ \dot{g}_3 \\ \dot{g}_4 \\ \vdots \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 & \cdots \\ 0 & 0 & 2 & 0 & \cdots \\ 0 & 0 & 0 & 3 & \cdots \\ 0 & 0 & 0 & 0 & \cdots \\ \vdots & \vdots & \vdots & \vdots & \ddots \end{bmatrix} \begin{bmatrix} g_1 \\ g_2 \\ g_3 \\ g_4 \\ \vdots \end{bmatrix} \quad \text{mit} \quad \begin{bmatrix} g_1 \\ g_2 \\ g_3 \\ g_4 \\ \vdots \end{bmatrix} = \begin{bmatrix} x \\ x^2 \\ x^3 \\ x^4 \\ \vdots \end{bmatrix}. \quad (3.12)$$

Eine endlichdimensionale Darstellung in Form von Matrizen ist nur möglich für den Fall, wenn die Observablen einen sogenannten *Koopman-invarianten Unterraum* aufspannen. Seien  $\alpha_1, \dots, \alpha_N, \beta_1, \dots, \beta_N \in \mathbb{R}$ . Dann ist der Unterraum  $\mathcal{M} = \text{span}\{g_1, g_2, \dots, g_N\}$  genau dann invariant gegenüber dem Koopman-Operator [27], falls alle Funktionen

$$g = \alpha_1 g_1 + \alpha_2 g_2 + \cdots + \alpha_N g_N \quad (3.13)$$

auch nach Anwendung des Koopman-Operators in diesem Unterraum liegen, d. h.

$$\mathcal{K}g = \beta_1 g_1 + \beta_2 g_2 + \cdots + \beta_N g_N. \quad (3.14)$$

Dieser Abschnitt stellt einen kurzen Einstieg in die theoretischen Grundlagen des Koopman-Operators dar. Im weiteren Verlauf liegt der Schwerpunkt auf seiner praktischen Anwendung. Für eine vertiefte Auseinandersetzung und zusätzliche Informationen zum Thema sei auf [27], [31] verwiesen.

### 3.1.2 Systemidentifikation mittels des Koopman-Operators

In Folgenden wird angenommen, dass sich die zu modellierende Dynamik der Strecke mit dem vollständig bekannten (=messbaren) Zustandsvektor  $\mathbf{x} \in \mathbb{R}^n$  und der Vektorfunktion  $\mathbf{f} : \mathbb{R}^n \rightarrow \mathbb{R}^n$  entweder als autonomes System

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}) \quad (3.15)$$

oder als eingangsaffines System

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}) + \mathbf{B}\mathbf{u} \quad (3.16)$$

mit der Eingangsgröße  $\mathbf{u} \in \mathbb{R}^p$  und der konstanten Eingangsmatrix  $\mathbf{B} \in \mathbb{R}^{n \times p}$  beschreiben lässt. Die verwendeten (Mess-)Daten dürfen sowohl von Simulationen als auch von realen Experimenten stammen und sind in Form von Snapshot-Matrizen

$$\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_{M-1}] \in \mathbb{R}^{n \times (M-1)}, \quad (3.17a)$$

$$\mathbf{U} = [\mathbf{u}_1, \mathbf{u}_2, \dots, \mathbf{u}_{M-1}] \in \mathbb{R}^{p \times (M-1)}, \quad (3.17b)$$

$$\mathbf{X}' = [\mathbf{x}_2, \mathbf{x}_3, \dots, \mathbf{x}_M] \in \mathbb{R}^{n \times (M-1)} \quad (3.17c)$$

anzuordnen, wobei  $\mathbf{x}_k = \mathbf{x}(k\Delta t)$  bzw.  $\mathbf{u}_k = \mathbf{u}(k\Delta t)$  mit  $k = 1, 2, \dots$ .  $\mathbf{X}'$  die aus  $\mathbf{X}$  um einen Zeitschritt  $\Delta t$  in die Zukunft verschobenen Zustände enthält. Die Daten müssen nicht zwingend von einer einzigen Messung oder Simulation stammen, es genügen Paare aus zwei aufeinander folgenden Snapshots  $\mathbf{x}_k$  und  $\mathbf{x}_{k+1}$ , die spaltenrichtig in  $\mathbf{X}$  und  $\mathbf{X}'$  eingetragen werden.

Im Folgenden werden Verfahren zur Snapshot-basierten Systemidentifikation erläutert: Dynamic Mode Decomposition (DMD) und Extended Dynamic Mode Decomposition (EDMD). Beide Verfahren lassen sich als endlichdimensionale Approximationen des Koopman-Operators interpretieren, weil der Zustandsübergang von einem Zeitschritt zum darauf folgenden jeweils durch eine Matrix approximiert wird.

### 3.1.3 Dynamic Mode Decomposition

Das DMD-Verfahren wurde 2008 von Peter Schmid erstmals im Kontext der Modellordnungsreduktion für hochdimensionale fluiddynamische Systeme formuliert [32], [33]. Der jeweilige Zustandsübergang von einem Zeitschritt in den folgenden Zeitschritt wird über die gesamte Datensequenz als lineare Dynamik angenommen.

Im Vergleich zur Proper Orthogonal Decomposition<sup>4</sup>, die auf einer Sortierung der Zustände nach räumlicher Korrelation und Energiegehalt beruht, wird beim DMD-Verfahren zusätzlich das zeitliche Verhalten der Zustände berücksichtigt<sup>5</sup> [38]. Das DMD-Verfahren approximiert eine zeitdiskrete Matrix  $\mathbf{A}_t$ , die den linearen Zusammenhang zwischen den Datenmatrizen  $\mathbf{X}$  und  $\mathbf{X}'$ , d. h. für jeden Zustandsübergang der Datensequenz

$$\mathbf{X}' \approx \mathbf{A}_t \mathbf{X} \quad (3.18)$$

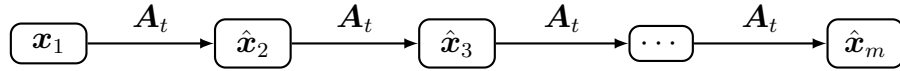
optimal beschreibt. Die kleinste-Quadrate-Lösung für  $\mathbf{A}_t$  ist gegeben durch

$$\mathbf{A}_t = \mathbf{X}' \mathbf{X}^+, \quad (3.19)$$

wobei  $\mathbf{X}^+$  die Pseudoinverse der Matrix  $\mathbf{X}$  ist [39]. Damit lässt sich das approximierte System linear im Zustandsraum mit der Dynamikmatrix  $\mathbf{A}_t$  beschreiben bzw. prädictieren, vgl. Abbildung 3.4,

$$\hat{\mathbf{x}}_{k+1} = \mathbf{A}_t \mathbf{x}_k. \quad (3.20)$$

Weil  $\mathbf{A}_t$  lediglich eine Approximation der betrachteten im Allgemeinen nichtlinearen Dynamik ist, handelt es sich bei  $\hat{\mathbf{x}}_{k+1}$  um eine Schätzung von  $\mathbf{x}_{k+1}$ . Die Matrix  $\mathbf{A}_t$  entspricht einer Approximation des Koopman-Operators  $\mathcal{K}_t$  für den Spezialfall, dass der Zustandsvektor als Observablenvektor aufgefasst wird, d. h. für  $\mathbf{g} : \mathbb{R}^n \rightarrow \mathbb{R}^n$ ,  $\mathbf{g}(\mathbf{x}) = \mathbf{x}$ .



**Abbildung 3.4** Beim DMD-Verfahren lässt sich die zeitliche Evolution der Zustände durch Multiplikation mit der Matrix  $\mathbf{A}_t$  prädictieren.

Das DMD-Verfahren lässt sich auf eingangsaffine Systeme (3.16) erweitern und wird dann auch als DMDc (Dynamic Mode Decomposition with Control) bezeichnet [40]. Dafür werden die Zustandsübergänge

$$\mathbf{X}' \approx \mathbf{A}_t \mathbf{X} + \mathbf{B}_t \mathbf{U} = [\mathbf{A}_t, \mathbf{B}_t] \begin{bmatrix} \mathbf{X} \\ \mathbf{U} \end{bmatrix} \quad (3.21)$$

betrachtet, sodass sich die kleinste-Quadrate-Lösung

$$[\mathbf{A}_t, \mathbf{B}_t] = \mathbf{X}' \begin{bmatrix} \mathbf{X} \\ \mathbf{U} \end{bmatrix}^+ \quad (3.22)$$

ergibt. Damit lässt sich das approximierte System linear prädictieren:

$$\hat{\mathbf{x}}_{k+1} = \mathbf{A}_t \mathbf{x}_k + \mathbf{B}_t \mathbf{u}_k. \quad (3.23)$$

<sup>4</sup>Die Proper Orthogonal Decomposition (POD) ist eine aus der Fluidodynamik stammende numerische Methode zur Analyse turbulenter Strömungen [34], die eng verwandt mit der Hauptkomponentenanalyse ist [35].

<sup>5</sup>Auch verwandt mit dem DMD-Verfahren ist der *Arnoldi-Algorithmus* [36], der Eigenwerte eines hochdimensionalen dynamischen Systems über Krylov-Unterräume approximiert, vgl. [37].



**Stabile DMD-Modelle** Im Bereich der Systemidentifikation existiert der Ansatz, identifizierte Modelle von eigentlich stabilen Systemen, die z. B. aufgrund von Messfehlern instabil geworden sind, nachträglich anzupassen, um ihre Stabilität und damit eine für die resultierende Dynamik elementare Systemeigenschaft sicherzustellen<sup>6</sup>. Ausgehend von einer instabilen Matrix  $\mathbf{A}_t \in \mathbb{R}^{n \times n}$  wird im Beitrag [42] die im Sinne der Frobeniusnorm nächste stabile Matrix

$$\inf_{\tilde{\mathbf{A}}_t \in \mathbb{S}_t^{n \times n}} \left\| \mathbf{A}_t - \tilde{\mathbf{A}}_t \right\|_F^2 \quad (3.24)$$

gesucht, wobei  $\mathbb{S}_t^{n \times n}$  die Menge aller zeitdiskreten stabilen  $n \times n$ -Matrizen ist<sup>7</sup>. Dafür wird eine neuartige Charakterisierung für zeitdiskrete Matrizen formuliert: eine Matrix  $\tilde{\mathbf{A}}_t$  ist genau dann stabil, wenn sie sich als

$$\tilde{\mathbf{A}}_t = \mathbf{S}^{-1} \mathbf{O} \mathbf{T} \mathbf{S} \quad (3.25)$$

formulieren lässt, wobei  $\mathbf{S}$  positiv definit,  $\mathbf{O}$  orthogonal und  $\mathbf{T}$  positiv semidefinit mit  $\|\mathbf{T}\| \leq 1$  sind. Die Matrix liegt dann in der sogenannten SOT-Form<sup>8</sup> vor. Damit lässt sich das nichtkonvexe Optimierungsproblem (3.24) in das gleichwertige Optimierungsproblem

$$\inf_{\tilde{\mathbf{A}}_t \in \mathbb{S}_t^{n \times n}} \left\| \mathbf{A}_t - \tilde{\mathbf{A}}_t \right\|_F^2 = \inf_{\mathbf{S} \succ 0, \mathbf{O} \in \mathcal{O}(n), \mathbf{T} \succeq 0, \|\mathbf{T}\| \leq 1} \left\| \mathbf{A}_t - \mathbf{S}^{-1} \mathbf{O} \mathbf{T} \mathbf{S} \right\|_F^2 \quad (3.26)$$

umformulieren, dessen zulässige Menge konvex ist. Die numerische Lösung erfolgt mittels eines projizierten Gradientenabstiegsverfahrens, vgl. [43]. Beim projizierten Gradientenabstieg wird versucht, eine gegebene Kostenfunktion unter einer festgelegten Beschränkung zu minimieren, indem jede Iteration zuerst einen Schritt in Richtung des negativen Gradienten und anschließend eine Projektion der Lösung in den erlaubten Suchraum berechnet. Die Ansätze aus [42] lassen sich auf datengetriebene DMD-Modelle anwenden.

Das DMD-Verfahren lässt sich ohne jegliches Vorwissen über die zu modellierende Dynamik anwenden. Diese Tatsache wird in der Literatur oft als Vorteil angeführt und aufgrund dieser Einfachheit wurde das DMD-Verfahren vielfach praktisch angewendet. Im Kontext hybrider Modelle erweist sich das rein datengetriebene Vorgehen jedoch als Nachteil, weil sich für die betrachteten mechatronischen Systeme meistens physikalisch motivierte Modelle herleiten lassen und dieses Vorwissen nutzbar gemacht werden soll. Außerdem adressiert das DMD-Verfahren vorwiegend fluiddynamische Fragestellungen, bei denen eine Modellordnungsreduktion, d. h. eine Transformation von einem hochdimensionalen in einen niedrigdimensionalen

<sup>6</sup>Darüber hinaus gibt es noch weitere strukturelle Anforderungen, die an datengetriebene Modelle gestellt werden können. Die Autor\*innen in [41] beispielsweise verallgemeinern diese Strategie als *Physics-informed DMD* für unterschiedliche gewünschte Matrixtypen mit jeweils korrespondierenden physikalischen Eigenschaften.

<sup>7</sup>Die Matrix  $\mathbf{A}_t$  ist stabil, wenn für alle Eigenwerte  $\lambda_1, \dots, \lambda_n$  von  $\mathbf{A}_t$  gilt  $|\lambda_i| \leq 1$  und es sich bei allen Eigenwerten auf dem Einheitskreis um halbeinfache Eigenwerte handelt [42].

<sup>8</sup>In der originalen Formulierung von [42] werden die Matrizen als  $\mathbf{S}$ ,  $\mathbf{U}$  und  $\mathbf{B}$  und die Schreibweise als SUB-Form bezeichnet. Dieses Dokument verwendet zur Vermeidung von Verwechslungen mit anderen Matrizen stattdessen die Bezeichnungen  $\mathbf{S}$ ,  $\mathbf{O}$  und  $\mathbf{T}$ .

Raum mit dem Ziel eines linearen Dynamikmodells, angestrebt wird. Im Gegensatz dazu erfordern komplexe mechatronische Fragestellungen aufgrund möglicher auftretender Nichtlinearitäten in der Regel eine höhere Modellkomplexität, d. h. die Transformation von einem typischerweise niedrigdimensionalen Raum in einen höherdimensionalen Raum mit dem Ziel eines linearen Dynamikmodells.

Das **DMD-Verfahren** eignet sich sowohl für autonome als auch nicht-autonome Systeme und besitzt aufgrund seiner extremen Einfachheit einen minimalen Rechenaufwand. Es erfordert Messdaten des **vollständigen** Zustandsvektors und lässt sich **ohne jegliches Vorwissen** über die zu modellierende Dynamik anwenden. Die resultierende zeitdiskrete lineare Systembeschreibung lässt sich geradlinig sowohl für die Analyse als auch den Regler- und Beobachterentwurf einsetzen.

### 3.1.4 Extended Dynamic Mode Decomposition

Das Verfahren Extended Dynamic Mode Decomposition (EDMD) verallgemeinert das DMD-Verfahren, weil die Observablen  $\mathbf{g}(\mathbf{x})$  zusätzlich zu den Zuständen nichtlineare Funktionen der Zustände enthalten. Dadurch entspricht dieses Verfahren einer endlichdimensionalen Approximation des Koopman-Operators. Im Folgenden werden der Algorithmus und die Prädiktionsvorschrift hergeleitet. Anschließend wird thematisiert, wie die Observablen zu konstruieren sind, wie sich die Koopman-Eigenzerlegung berechnet und wie sich das Verfahren um einen Systemeingang erweitern lässt.

#### Algorithmus

Für das EDMD-Verfahren werden  $N$  Observablen

$$\Psi : \mathbb{R}^n \rightarrow \mathbb{R}^N, \quad \Psi(\mathbf{x}) = [\psi_1(\mathbf{x}), \psi_2(\mathbf{x}), \dots, \psi_N(\mathbf{x})]^\top \quad (3.27)$$

definiert<sup>9</sup>, wobei in den meisten Fällen  $N > n$  gilt. Um möglichst vielfältige Dynamiken zu modellieren, bietet es sich im Allgemeinen an, Polynome, radiale Basisfunktionen oder trigonometrische Funktionen für  $\Psi(\mathbf{x})$  zu verwenden [44]. Eine beliebige skalare Funktion  $\theta : \mathbb{R} \rightarrow \mathbb{R}$  lässt sich als Linearkombination

$$\theta = \sum_{k=1}^N a_k \psi_k = \mathbf{a} \Psi, \quad a_k \in \mathbb{R} \quad \text{mit} \quad \mathbf{a} \in \mathbb{R}^{1 \times N} \quad (3.28)$$

der Funktionen  $\Psi(\mathbf{x})$  beschreiben. Für die zeitliche Evolution von  $\theta$  gilt mit dem Koopman-Operator

$$\mathcal{K}_t \theta = \mathcal{K}_t(\mathbf{a} \Psi) = \mathbf{a} \Psi \circ \mathbf{F}_t. \quad (3.29)$$

Die endlichdimensionale Approximation von  $\mathcal{K}_t$  durch  $\mathbf{K}_t \in \mathbb{R}^{N \times N}$

$$\mathbf{a} \Psi \circ \mathbf{F}_t = \mathbf{a} \mathbf{K}_t \Psi + r \quad \Rightarrow \quad r = \mathbf{a} (\Psi \circ \mathbf{F}_t - \mathbf{K}_t \Psi) \quad (3.30)$$

<sup>9</sup>[44] formuliert  $\Psi(\mathbf{x})$  in transponierter Darstellung, d. h. als Zeilenvektor.

resultiert in einem Fehler  $r$ , der sich durch Anpassung von  $K_t$  über Messdaten des betrachteten Systems minimieren lässt. Über alle Messpunkte wird die Kostenfunktion

$$L = \frac{1}{2} \sum_{m=1}^{M-1} \|r(\mathbf{x}_m)\|^2 = \frac{1}{2} \sum_{m=1}^{M-1} \|\mathbf{a}((\Psi \circ \mathbf{F}_t)(\mathbf{x}_m) - \mathbf{K}_t \Psi(\mathbf{x}_m))\|^2 \quad (3.31)$$

definiert. Mit  $(\Psi \circ \mathbf{F}_t)(\mathbf{x}_m) = \Psi(\mathbf{x}_{m+1})$  folgt

$$L = \frac{1}{2} \sum_{m=1}^{M-1} \|\mathbf{a}(\Psi(\mathbf{x}_{m+1}) - \mathbf{K}_t \Psi(\mathbf{x}_m))\|^2. \quad (3.32)$$

Die Minimierung von Gleichung (3.32) entspricht einem kleinste-Quadrate-Problem [44] mit dem globalen Optimum [45] in

$$\mathbf{K}_t = \mathcal{G}^+ \mathcal{A} \quad (3.33)$$

mit

$$\mathcal{G} = \frac{1}{M} \sum_{m=1}^{M-1} \Psi(\mathbf{x}_m) \Psi^\top(\mathbf{x}_m) \in \mathbb{R}^{N \times N}, \quad (3.34)$$

$$\mathcal{A} = \frac{1}{M} \sum_{m=1}^{M-1} \Psi(\mathbf{x}_m) \Psi^\top(\mathbf{x}_{m+1}) \in \mathbb{R}^{N \times N}. \quad (3.35)$$

Eine alternative Berechnungsvorschrift für  $\mathbf{K}_t$  lässt sich analog zum DMD-Algorithmus mittels der erweiterten Snapshotmatrizen formulieren [46]. Dafür werden die Observablen für  $\mathbf{X}$  und  $\mathbf{X}'$  ausgewertet

$$\Psi(\mathbf{X}) = [\Psi(\mathbf{x}_1), \Psi(\mathbf{x}_2), \dots, \Psi(\mathbf{x}_{M-1})] \in \mathbb{R}^{N \times (M-1)}, \quad (3.36a)$$

$$\Psi(\mathbf{X}') = [\Psi(\mathbf{x}_2), \Psi(\mathbf{x}_3), \dots, \Psi(\mathbf{x}_M)] \in \mathbb{R}^{N \times (M-1)}, \quad (3.36b)$$

sodass sich  $\mathbf{K}_t$  über den Zusammenhang

$$\Psi(\mathbf{X}') \approx \mathbf{K}_t \Psi(\mathbf{X}) \quad (3.37)$$

mittels der Pseudoinversen berechnen lässt

$$\mathbf{K}_t = \Psi(\mathbf{X}') \Psi^+(\mathbf{X}). \quad (3.38)$$

Beide Berechnungsalgorithmen sind äquivalent zueinander, eine Herleitung dazu findet sich in [45]. Die resultierende Systembeschreibung des EDMD-Modells, d. h. die zeitliche Evolution der Observablen  $\Psi$  für einen Prädiktionsschritt ist durch

$$\hat{\Psi}(\mathbf{x}_{k+1}) = \mathbf{K}_t \Psi(\mathbf{x}_k), \quad (3.39)$$

gegeben. Die Prädiktion der Originalzustände erfordert eine Rücktransformation

$$\mathbf{x} = \mathbf{P} \Psi(\mathbf{x}), \quad (3.40)$$

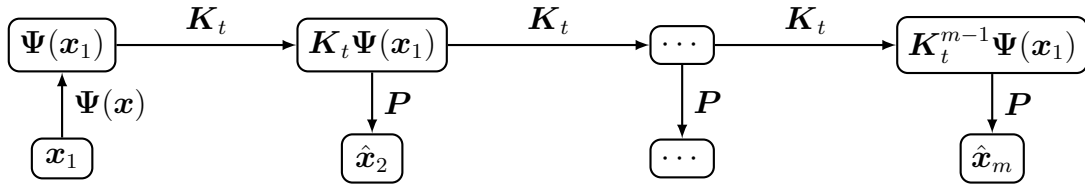
vgl. Abbildung 3.5. Falls die Observablen den Zustandsvektor als Identität an erster Stelle beinhalten, d. h.  $\Psi(\mathbf{x}) = [\mathbf{x}, \psi_{n+1}(\mathbf{x}), \dots, \psi_N(\mathbf{x})]^\top$ , reduziert sich die Lösung auf

$$\mathbf{P} = [\mathbf{I}_n, \mathbf{0}_{n \times (N-n)}] \in \mathbb{R}^{n \times N}, \quad (3.41)$$

wobei  $\mathbf{I}_n$  die  $n \times n$ -Einheitsmatrix und  $\mathbf{0}_{n \times (N-n)}$  die  $n \times (N - n)$ -Nullmatrix sind. Andernfalls wird sie numerisch über

$$\mathbf{P} = \mathbf{X} \Psi(\mathbf{X}') \quad (3.42)$$

approximiert [47]. Das Vorgehen bei der Prädiktion mittels des EDMD-Verfahrens ist in der Abbildung 3.5 dargestellt.



**Abbildung 3.5** Resultierende Systembeschreibung beim EDMD-Verfahren. Die Observablen  $\Psi(\mathbf{x})$  weisen eine höhere Dimension als der Zustand  $\mathbf{x}$  auf, daher wird  $\Psi(\mathbf{x})$  auch als *Lifting-Operator* bezeichnet.

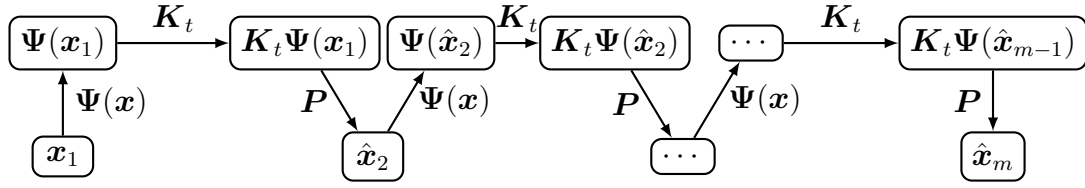
### EDMD-Prädiktion mit Korrektur

Für die meisten nichtlinearen dynamischen Systeme lassen sich nicht ohne Weiteres Observablen finden, die einen Koopman-invarianten Unterraum aufspannen. Das bedeutet, dass sich nicht alle Zustände  $\Psi(\mathbf{x}_{k+1})$  durch Linearkombination aus  $\Psi(\mathbf{x}_k)$  exakt präzisieren lassen. Dadurch resultiert ein Fehler, der mit jedem Prädiktionsschritt weiter anwächst [48]. Dieser Fehler kann signifikant reduziert werden, indem die Observablen  $\Psi(\mathbf{x})$  in jedem Prädiktionsschritt nach Extraktion des Zustandsvektors  $\mathbf{x}$  mittels Gleichung (3.40) neu ausgewertet werden. Dadurch müssen nicht mehr alle Observablen durch die Anwendung von  $\mathbf{K}_t$  prädictiert werden, sodass die Forderung eines Koopman-invarianten Unterraums entfällt. Die resultierende Prädiktionsvorschrift

$$\hat{\mathbf{x}}_{k+1} = \mathbf{P} \hat{\Psi}(\mathbf{x}_{k+1}) = \mathbf{P} \mathbf{K}_t \Psi(\mathbf{x}_k) \quad (3.43)$$

wurde erstmalig in [25] als *EDMD-Prädiktion mit Korrektur* vorgestellt<sup>10</sup> und ist in der Abbildung 3.6 als alternierender Prozess zwischen dem Originalzustand und den höherdimensionalen Observablen dargestellt. Die EDMD-Prädiktion mit Korrektur ist keine tatsächlich lineare Systembeschreibung, weil in jedem Berechnungsschritt zusätzlich zur Matrix-Vektor-Multiplikation die Auswertung einer nichtlinearen Funktion erforderlich ist. Dadurch erhöht sich der Rechenaufwand bei der Prädiktion im Vergleich zum reinen EDMD-Verfahren.

<sup>10</sup>Ein ähnlicher Ansatz wurde in [49] formalisiert, indem zwischen dem *Domain*- und *Output*-Raum unterschieden wird, wobei der Domain-Raum hier  $\Psi(\mathbf{x})$  und der Output-Raum hier  $\mathbf{x}$  entspricht.



**Abbildung 3.6** Prädiktionsvorschrift bei der EDMD-Prädiktion mit Korrektur. Die Prädiktion von  $\Psi$  mittels  $K_t$  erfolgt im höherdimensionalen Raum, jedoch wird der Zustand  $x$  in jedem Prädiktionsschritt mittels  $P$  extrahiert.

### Wahl der Observablen

Die Wahl der Observablen hat einen entscheidenden Einfluss auf die Genauigkeit der approximierten Systembeschreibung und sollte in Abhängigkeit der zugrunde liegenden Dynamik und den verwendeten Daten erfolgen. Dabei gibt es unterschiedliche Ansätze. [44] empfehlen beispielsweise die Verwendung von Hermiteischen Polynomen oder radialen Basisfunktionen [50]. [51] nutzen den Kernel-Trick<sup>11</sup> zur impliziten Berechnung der Observablen für den Fall, dass die Anzahl der Messpunkte deutlich geringer als die Anzahl der angestrebten Observablen ist. Aufgrund verfügbarer Messeinrichtungen lässt sich dieses Szenario für Zeitreihen von regelungstechnischen Prüfständen ausschließen. Ein weit verbreiteter Ansatz bestimmt die Observablen implizit mittels eines (tiefen) neuronalen Netzes. Neben zahlreichen weiteren Veröffentlichungen sei hier vor allem [52] genannt. In dieser Arbeit wird im Kontext regelungstechnischer Systeme ein gewisses Maß an physikalischem Vorwissen über die Systemdynamik angenommen. Daher ist es wünschenswert, dieses Vorwissen möglichst sinnvoll beim Design der Observablen zu berücksichtigen<sup>12</sup>. Eine hierfür geeignete Strategie für die Wahl der Observablen ist motiviert durch die Taylorreihenentwicklung [54]. Demnach empfiehlt es sich, die (mehrfachen) zeitlichen Ableitungen der in der betrachteten Dynamik vorkommenden Funktionen in  $\Psi(x)$  zu berücksichtigen [45], [55]. Für den Fall, dass die Dynamik mehrere Nichtlinearitäten enthält, ist es aufgrund möglicher Parameterunsicherheiten empfehlenswert, jeden Term einzeln zu berücksichtigen [56].

### Erweiterung um einen Systemeingang

Beim EDMD-Verfahren gibt es unterschiedliche Ansätze für die Erweiterung des Modells um einen Steuereingang. Im Allgemeinen wird der Koopman-Zustandsvektor um die Eingänge bzw. nichtlineare Funktionen der Eingänge erweitert [45], [49], [55]. Für den Fall eingangsaffiner Systeme (3.16) lässt sich ein ähnlicher Ansatz verfolgen wie beim DMDc-Verfahren. Dafür wird der Zusammenhang

$$\Psi(X') \approx K_t \Psi(X) + B_t U = [K_t, B_t] \begin{bmatrix} \Psi(X) \\ U \end{bmatrix} \quad (3.44)$$

<sup>11</sup>Der Kernel-Trick ist eine etablierte Technik aus dem Bereich der Mustererkennung. Die betrachteten Daten werden über eine Kernel-Funktion in einen höherdimensionalen Raum transformiert [10].

<sup>12</sup>[53] beschreibt dieses Konzept als *parenting in learning*, d. h. dass vorhandenes Expert\*innenwissen gezielt dem Lernvorgang vermittelt wird.

betrachtet, sodass sich die kleinste-Quadrate-Lösung

$$[\mathbf{K}_t, \mathbf{B}_t] = \Psi(\mathbf{X}') \begin{bmatrix} \Psi(\mathbf{X}) \\ \mathbf{U} \end{bmatrix}^+ \quad (3.45)$$

ergibt. Anschließend lässt sich das approximierte System

$$\hat{\Psi}(\mathbf{x}_{k+1}) = \mathbf{K}_t \Psi(\mathbf{x}_k) + \mathbf{B}_t \mathbf{u}_k \quad (3.46)$$

prädictieren. Dieses Verfahren wird auch als EDMDc bezeichnet und lässt sich wie das EDMD-Verfahren um einen Korrekturansatz erweitern, vgl. Gleichung (3.43)

$$\hat{\mathbf{x}}_{k+1} = \mathbf{P} \hat{\Psi}(\mathbf{x}_{k+1}) = \mathbf{P} (\mathbf{K}_t \Psi(\mathbf{x}_k) + \mathbf{B}_t \mathbf{u}_k). \quad (3.47)$$

Eine weitere Strategie zur Autonomisierung eines aktuierten Systems wird in [57], [58], [59], [60], [61], [62] beschrieben. Die möglichen Systemeingänge  $\mathbf{u} \in \mathcal{U}$  werden in endlich viele Werte  $\{\mathbf{u}^1, \dots, \mathbf{u}^m\} \in \mathcal{U}$  diskretisiert. Dadurch wird das aktuierte System durch endlich viele autonome Systeme mit jeweils konstantem Eingang ersetzt. Um nicht eingangsaffine Systeme zu lernen, kann der Eingang durch ein neuronales Netz kodiert werden [63].

### Stabile EDMD-Modelle

Ähnlich wie DMD-Modelle lassen auch EDMD-Modelle nachträglich so anpassen, dass sie stabil sind. Die Autor\*innen in [48] leiten Fehlerschranken von EDMD-Modellen her und plädieren dafür, zur Erhöhung der Prädiktionsgüte des reinen EDMD-Verfahrens in allen Fällen<sup>13</sup> einen stabilen Koopman-Operator zu verwenden. In Anlehnung an stabile DMD-Modelle entspricht die Bestimmung des im Sinne der Frobeniusnorm nächsten stabilen Koopman-Operators  $\tilde{\mathbf{K}}_t$  zu einem anhand von Messdaten approximierten (möglicherweise instabilen) Koopman-Operator  $\mathbf{K}_t$  dem Optimierungsproblem

$$\inf_{\tilde{\mathbf{K}}_t \in \mathbb{S}^{N \times N}} \|\mathbf{K}_t - \tilde{\mathbf{K}}_t\|_F^2 = \inf_{\mathbf{S} \succ 0, \mathbf{O} \in \mathcal{O}(N), T \succeq 0, \|T\| \leq 1} \|\mathbf{K}_t - \mathbf{S}^{-1} \mathbf{O} \mathbf{T} \mathbf{S}\|_F^2. \quad (3.48)$$

Alternativ schlagen die Autor\*innen in [48], [64] vor, stattdessen die nächste stabile Matrix zur Kleinste-Quadrate-Lösung (3.38) zu betrachten, d. h. die verwendeten Daten  $\mathbf{X}, \mathbf{X}'$  (vgl. Gleichung (3.17)) mit zu berücksichtigen

$$\begin{aligned} & \inf_{\tilde{\mathbf{K}}_t \in \mathbb{S}^{N \times N}} \|\Psi(\mathbf{X}') - \tilde{\mathbf{K}}_t \Psi(\mathbf{X})\|_F^2 \\ &= \inf_{\mathbf{S} \succ 0, \mathbf{O} \in \mathcal{O}(N), T \succeq 0, \|T\| \leq 1} \|\Psi(\mathbf{X}') - \mathbf{S}^{-1} \mathbf{O} \mathbf{T} \mathbf{S} \Psi(\mathbf{X})\|_F^2. \end{aligned} \quad (3.49)$$

Diese Formulierung (3.49) führt zu einer anderen Lösung als Gleichung (3.48), wobei eine erhöhte Genauigkeit des resultierenden stabilen EDMD-Modells bezüglich der verwendeten Messdaten erzielt wird. Außerdem zeichnet sich dieser Ansatz dadurch aus, dass bereits in der ersten Iteration ein stabiles Modell berechnet wird. Durch

<sup>13</sup>Sogar für den Fall, dass das zugrundeliegende System eine instabile Dynamik aufweist [48].

die Verwendung der Daten  $\mathbf{X}, \mathbf{X}', \mathbf{U}$  wird der Abstand zur Kleinste-Quadrate-Lösung (3.38) mit jeder Iteration reduziert. Diese Eigenschaft ist vorteilhaft für Online-Anwendungen, bei denen das stabile Modell schnell berechnet werden muss. Die iterative Berechnung von  $\mathbf{B}_t$  erweitert den Suchraum im Vergleich für den Fall, in dem nur  $\mathbf{K}_t$  angepasst wird.

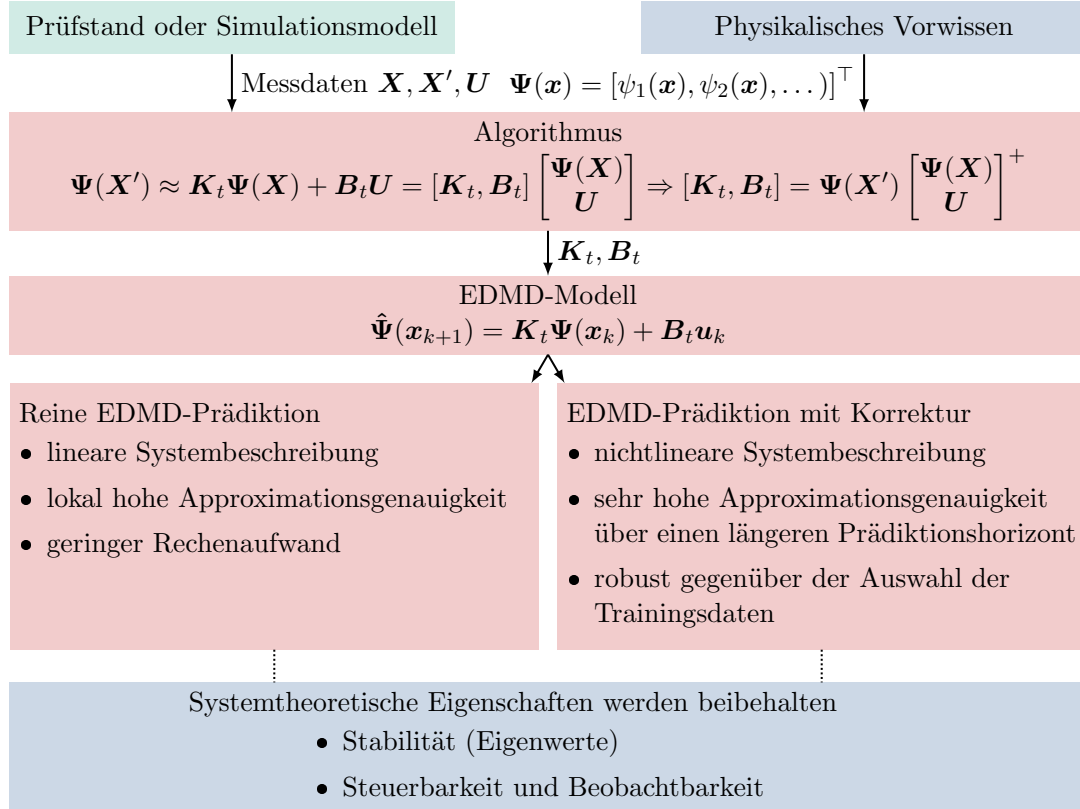
Das **EDMD-Verfahren** approximiert autonome und nicht-autonome nichtlineare Dynamiken anhand von (Mess-)Daten sowie physikalischem Vorwissen über die Form der auftretenden Nichtlinearitäten als **lineare zeitdiskrete Modelle**. Es erfordert Messdaten des **vollständigen** Zustandsvektors. Die reine EDMD-Prädiktion liefert eine hohe Genauigkeit für einen kurzen Horizont, wohingegen die EDMD-Prädiktion mit Korrektur auch über einen längeren Zeithorizont eine sehr hohe Genauigkeit liefert. Hinsichtlich der Stabilität sowie der Steuerbarkeit und Beobachtbarkeit lässt sich anhand von Beispielanwendungen eine Korrespondenz des zugrunde liegenden nichtlinearen Systems und dem jeweiligen approximierten EDMD-Modell feststellen, vgl. [25]. Dieses Ergebnis lässt sich nicht verallgemeinern, sodass die Steuerbarkeit bzw. Beobachtbarkeit im jeweiligen Einzelfall zu prüfen ist, vgl. [16]. Die Abbildung 3.7 fasst die Ergebnisse dieses Abschnitts zusammen. Die Analyse des Rechenaufwands bei der Prädiktion anhand der Betrachtung von Taktzyklen<sup>a</sup> zeigt, dass die beiden EDMD-Prädiktionsverfahren als datengetriebene Alternativen zu klassischen numerischen Integrationsmethoden verwendet werden können, vgl. studentische Arbeit [67]. Die reine EDMD-Prädiktion eignet sich aufgrund ihrer hohen Genauigkeit und einem gleichzeitig geringen Rechenaufwand bei kurzen Zeitintervallen besonders für Anwendungen mit Echtzeitanforderungen<sup>b</sup>, wie z. B. MPC.

<sup>a</sup>Ein Taktzyklus ist in der Computertechnik der benötigte Zeitraum für die Ausführung einer elementaren Rechenoperation auf einem Prozessor [65]. Der Rechenaufwand einer Gleitkommaoperation lässt sich durch die Anzahl benötigter Taktzyklen angeben. Das Handbuch [66] bietet eine überaus umfangreiche Auseinandersetzung mit der Bestimmung von Taktzyklen für unterschiedliche Gleitkommaoperationen und Prozessorarchitekturen.

<sup>b</sup>Echtzeitfähige Systeme sind in der Lage, das Ergebnis einer Berechnung innerhalb eines vorher fest definierten Zeitintervalls garantiert berechnen zu können [68].

## Beispielhafte Anwendung

Aufgrund der linearen Systembeschreibung sowie der Möglichkeit, vorhandenes physikalisches Vorwissen auszunutzen, scheint das EDMD-Verfahren eine vielversprechende Strategie für die Entwicklung regelungstechnisch verwertbarer Streckenmodelle zu sein. Daher wird im Folgenden die Anwendbarkeit des EDMD-Verfahrens beispielhaft hinsichtlich der erreichbaren Prädiktionsgüte demonstriert. Hierfür wird das numerisch simulierte nichtlineare Verhalten des jeweiligen Originalsystems mit dem approximierten EDMD-Modell verglichen. Als akademisches Beispiel wird das im Abschnitt 2.1 eingeführte Einfachpendel betrachtet, das mehrere isolierte Ruhelagen aufweist. Als experimentelles Beispiel wird der Schlagmechanismus des Golfroboters betrachtet, dessen Dynamik nichtlineare Reibungseffekte enthält.



**Abbildung 3.7** Das EDMD-Verfahren bietet die Möglichkeit anhand von (Mess-) Daten und physikalischem Vorwissen ein lineares zeitdiskretes Modell zu approximieren, wobei die Approximationsgenauigkeit von den gewählten Observablen, den Trainingsdaten und dem Prädiktionsverfahren abhängt. Bezüglich systemtheoretischer Eigenschaften konnte für die untersuchten Beispielsysteme eine Korrespondenz zwischen dem zugrunde liegenden Originalsystem und dem EDMD-approximierten Modell festgestellt werden.

**Pendel** Die Dynamik des im Gelenk aktuierten nichtlinearen Pendels, vgl. Abschnitt 2.1, wird durch

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} x_2 \\ -\frac{g}{l} \sin x_1 - \frac{d}{ml^2} x_2 + \frac{1}{ml^2} u \end{bmatrix}, \quad (3.50a)$$

$$y = x_1 \quad (3.50b)$$

mit  $m = 1 \text{ kg}$ ,  $g = 9,81 \text{ m s}^{-2}$ ,  $l = 0,5 \text{ m}$ ,  $d = 0,1 \text{ kg m}^2 \text{ s}^{-1}$  beschrieben. Das Pendel weist ein kontinuierliches Spektrum auf, was die Approximation mittels eines EDMD-Modells erschwert. Im Gegensatz zu der studentischen Arbeit [69], die zur Abbildung der unterschiedlichen Eigenfrequenzen eine Schar winkelabhängiger EDMD-Modelle präsentiert, verfolgt der hier präsentierte Ansatz das Ziel, die Pendeldynamik mittels eines einzigen Modells zu beschreiben.

Die Prädiktionsgenauigkeit des approximierten EDMD-Modells hängt stark von der sorgfältigen Auswahl der Observablen und den gewählten Trainingsdaten ab, wie im Beitrag [25] erläutert ist. Die Observablen werden so gewählt, dass sie den Zustandsvektor sowie mehrmalige zeitliche Ableitungen der nichtlinearen Terme des jeweiligen Originalsystems enthalten. Für das Pendel ergeben sich aufgrund des Terms

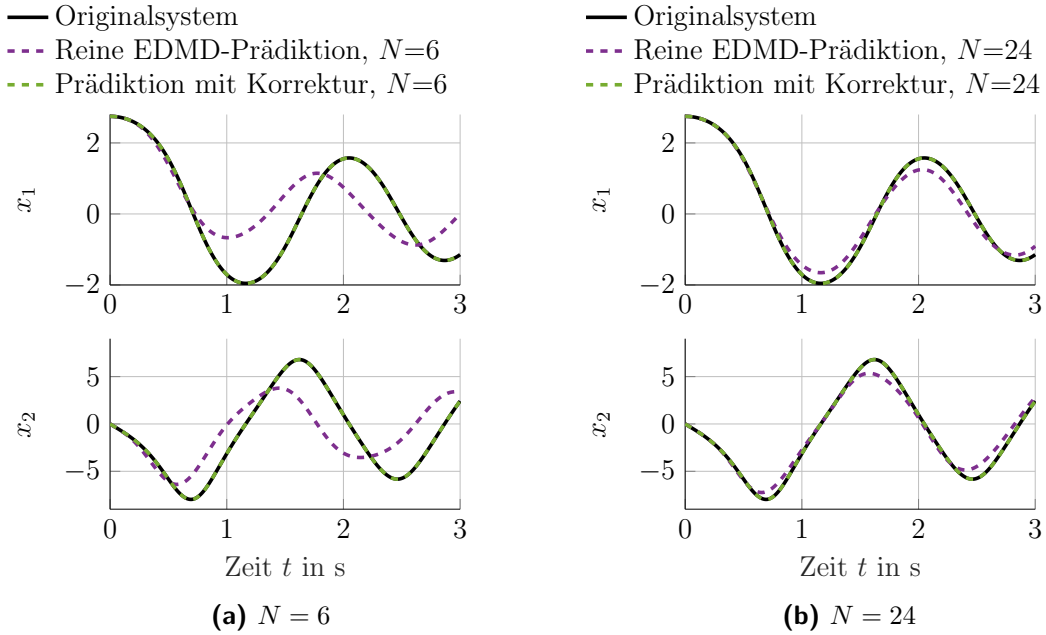


$\sin x_1$  unter Berücksichtigung der Kettenregel die Observablen

$$\begin{aligned} \Psi_{\text{Pendel}}(\mathbf{x}) = & \left[ x_1, x_2, \sin x_1, x_2 \cos x_1, x_2^2 \sin x_1, \sin x_1 \cos x_1, x_2 \cos^2 x_1, \dots \right. \\ & x_2 \sin^2 x_1, x_2^3 \cos x_1, \sin^3 x_1, \sin x_1 \cos^2 x_1, x_2^4 \sin x_1, \dots \\ & x_2^2 \sin x_1 \cos x_1, x_2^3 \sin^2 x_1, x_2^3 \cos^2 x_1, x_2 \cos^3 x_1, x_2^5 \cos x_1, \dots \\ & x_2 \sin^2 x_1 \cos x_1, x_2^6 \sin x_1, x_2^2 \sin^3 x_1, \sin^3 x_1 \cos x_1, \dots \\ & \left. \sin x_1 \cos^3 x_1, x_2^4 \sin x_1 \cos x_1, x_2^2 \sin x_1 \cos^2 x_1, \dots \right]^\top. \end{aligned} \quad (3.51)$$

Für die Erzeugung der Trainingsdaten werden jeweils 100 simulierte Trajektorien mit einer Dauer von jeweils 3 s, einer Schrittweite von  $\Delta t = 0,01$  s und zufälligen Anfangsauslenkungen mit  $u = 0$  verwendet. Für das Pendel sollen die Anfangsauslenkungen im Einzugsbereich der unteren Ruhelage  $[0, 0]^\top$  liegen.

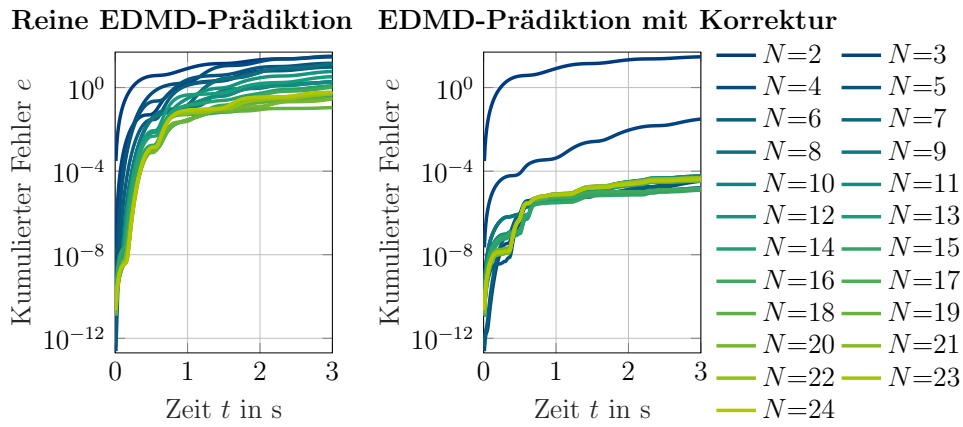
Die Prädiktionsgenauigkeit der EDMD-Verfahren lässt sich anschließend anhand von Testtrajektorien beurteilen. Als Anfangsauslenkung wird  $\mathbf{x}_0 = [7\pi/8, 0]^\top$  gewählt, um zu validieren, mit welcher Genauigkeit die nichtlinearen Anteile der Dynamik abgebildet werden. Die Abbildung 3.8 zeigt die Prädiktionsgenauigkeit der EDMD-Prädiktionen in Abhängigkeit der Anzahl der verwendeten Observablen (3.51) im Vergleich zur numerischen Simulation des nichtlinearen Originalsystems. Die EDMD-Prädiktion mit Korrektur liefert eine sehr hohe Genauigkeit, während die reine EDMD-Prädiktion die Trajektorie lediglich lokal approximiert. Diese Beobachtung lässt sich dadurch erklären, dass die festgelegten Observablen keinen Koopman-invarianten Unterraum aufspannen. Daher können nicht alle Zustände der Observablen durch Linearkombination der vorigen Zustände korrekt prädiziert werden. Durch eine Erhöhung der Observablenanzahl weicht die Prädiktion zeitlich später ab. Die Abbildung 3.9 zeigt den kumulierten mittleren quadratischen Fehler



**Abbildung 3.8** Beispielhafte EDMD-Prädiktion des Pendels für eine unterschiedliche Anzahl an Observablen, vgl. Gleichung (3.51). Die Zustände  $x_1$  und  $x_2$  beschreiben den Winkel und die Winkelgeschwindigkeit des Pendels.

$$e(t_k) = \sum_{m=1}^k \frac{1}{k} \|\mathbf{x}_{\text{Originalsystem}}(t_m) - \mathbf{x}_{\text{Prädiktion}}(t_m)\|^2 \quad (3.52)$$

in Abhängigkeit der Anzahl der Observablen (3.51) für das Pendel. Hierbei fällt auf, dass eine Erhöhung der Observablenanzahl  $N$  in der hier vorgeschlagenen Form nicht zwangsläufig zu einer beliebig starken Verringerung des Fehlers führt. Es ist außerdem ersichtlich, dass bereits ab  $N = 4$  die EDMD-Prädiktion mit Korrektur den minimalen Fehler erreicht. Das bedeutet, dass eine Erhöhung der Observablenanzahl durch zusätzliche zeitliche Ableitungen keine weiteren Vorteile zeigt. Es ist wichtig anzumerken, dass der Fehler bereits für  $N \geq 3$  ausgesprochen klein ist und die logarithmische Darstellung bei der Betrachtung dazu verleitet, den Fehler zu überschätzen.



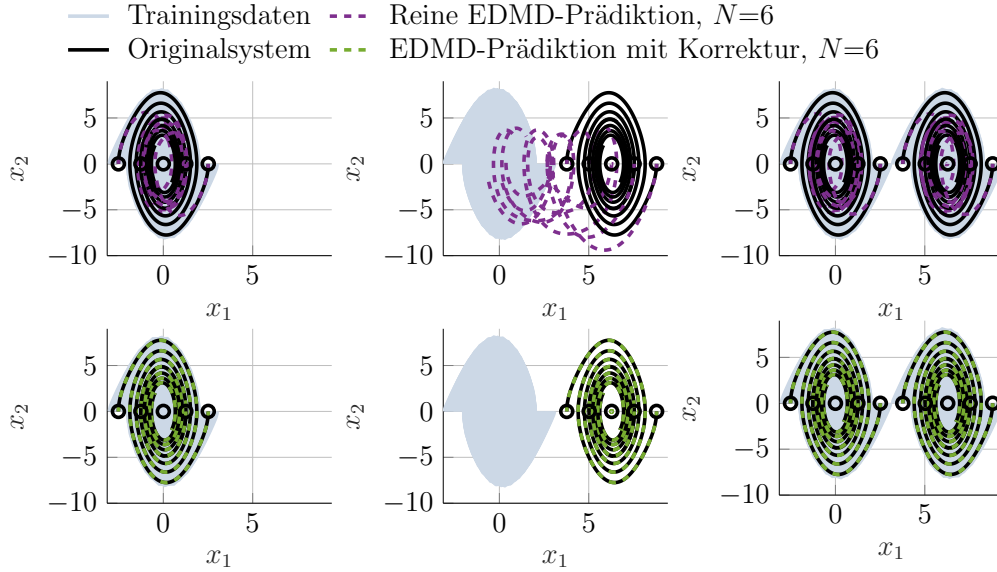
**Abbildung 3.9** Einfluss der Anzahl der verwendeten Observablen auf die Prädiktionsgenauigkeit am Beispiel des Pendels für dieselbe Testtrajektorie wie in der Abbildung 3.8.

Darüber hinaus hat die Wahl der Trainingsdaten einen Einfluss auf die erreichbare Prädiktionsgenauigkeit der Testtrajektorien. Dieser Zusammenhang ist in der Abbildung 3.10 am Beispiel des Pendels veranschaulicht. Für den Fall, dass die Testtrajektorien in einem Bereich des Zustandsraums verlaufen, der die Trainingsdaten umfasst, erhöht sich die Prädiktionsgenauigkeit des reinen EDMD-Verfahrens. Im Gegensatz dazu liefert das EDMD-Verfahren mit Korrektur eine extrem hohe Genauigkeit über einen langen Prädiktionshorizont bei gleichzeitiger Robustheit gegenüber der Auswahl der Trainingsdaten.

**Schlagmechanismus des Golfroboters** Der Schlagmechanismus des Golfroboters, vgl. Abschnitt 6.1, zeichnet sich durch ein nichtlineares Reibungsmoment aus, sodass es sich empfiehlt, die EDMD-Variablen im Rahmen der Verwendung von Vorwissen folgendermaßen zu wählen

$$\Psi_{\text{Golf}}(\mathbf{x}) = [x_1, x_2, \sin x_1, \operatorname{sgn} x_2 |mx_2^2 a + mg \cos x_1|]^\top. \quad (3.53)$$

Für die Erzeugung der Trainingsdaten wurde der Schlagmechanismus mit unterschiedlichen Verläufen für  $u$  (Chirp-, Sinus- und Sprungsignale mit variierender Amplitude und Frequenz, vgl. Abbildung 3.11) angeregt und der Systemausgang  $y = x_1$



**Abbildung 3.10** Einfluss der Trainingsdaten auf die Prädiktionsgenauigkeit am Beispiel des Pendels. Von links nach rechts werden unterschiedliche Kombinationen aus Trainingsdaten und Testtrajektorien im Phasenporträt betrachtet. Die Anfangspunkte der Testtrajektorien sind jeweils mit einem schwarzen Kreis gekennzeichnet.

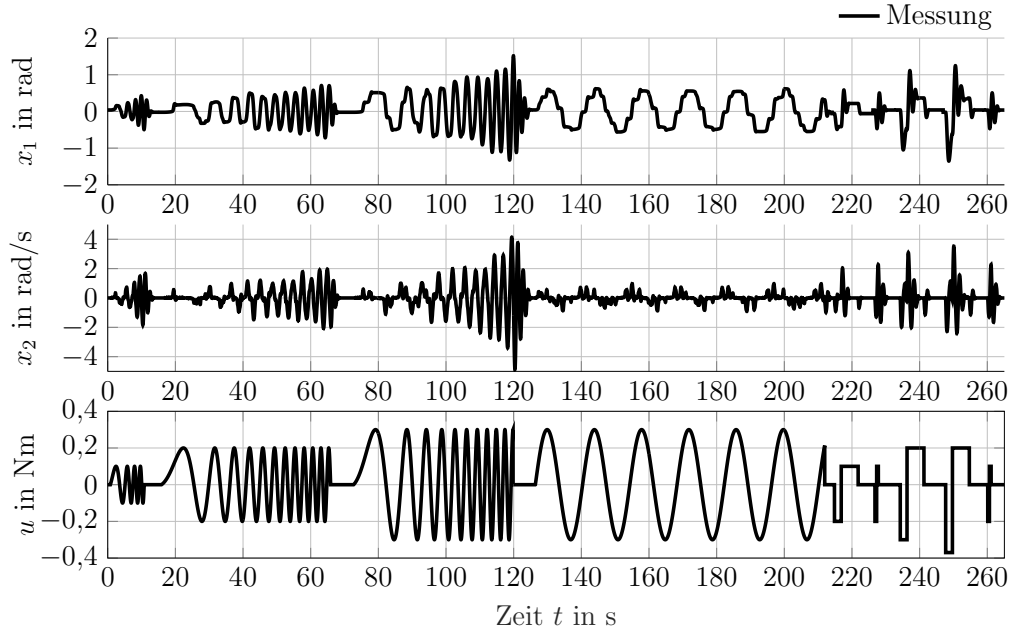
mit einer Abtastrate von 1 kHz gemessen. Der Verlauf für  $x_2$  wurde anschließend offline durch numerische Differentiation einer durch Spline-Kurven interpolierten Approximation des Verlaufs von  $x_1$  geschätzt. Zur Beurteilung der Prädiktionsgenauigkeit wird anschließend eine Testanregung untersucht, die nicht in den Trainingsdaten enthalten ist. Die Abbildung 3.12 zeigt die Prädiktionsgenauigkeit der EDMD-Verfahren im Vergleich zum nichtlinearen physikalischen Modell (6.11). Anhand der Zeitverläufe und dem kumulierten mittleren quadratischen Fehler

$$e(t_k) = \sum_{m=1}^k \frac{1}{k} (x_{1,\text{Messung}}(t_m) - x_{1,\text{Prädiktion}}(t_m))^2 \quad (3.54)$$

lässt sich erkennen, dass die reine EDMD-Prädiktion eine geringfügig schwächere Genauigkeit als das nichtlineare physikalische Modell liefert. Im Gegensatz dazu ist die EDMD-Prädiktion mit Korrektur jedoch in der Lage, die nichtlineare Dämpfung nach ca. 8 s präziser abzubilden, wodurch sich der kumulierte Fehler signifikant verringert.

### Automatische Modellaktualisierung

Eine Herausforderung mechatronischer Systeme liegt in vorhersehbaren oder unvorhersehbaren Systemveränderungen, z. B. die Installation eines neuen Bauteils, Verschleiß oder Temperaturschwankungen während des Betriebs. Um dennoch eine gleichbleibend hohe Regelungsgüte gewährleisten zu können, muss die Informationsverarbeitung in der Lage sein, diese Änderungen zu erfassen und sich automatisch anzupassen. Daher wurde in [70] eine Erweiterung um selbstlernende Streckenmodelle entwickelt, die sich auf Basis laufend aufgezeichneter Messdaten an veränderliches

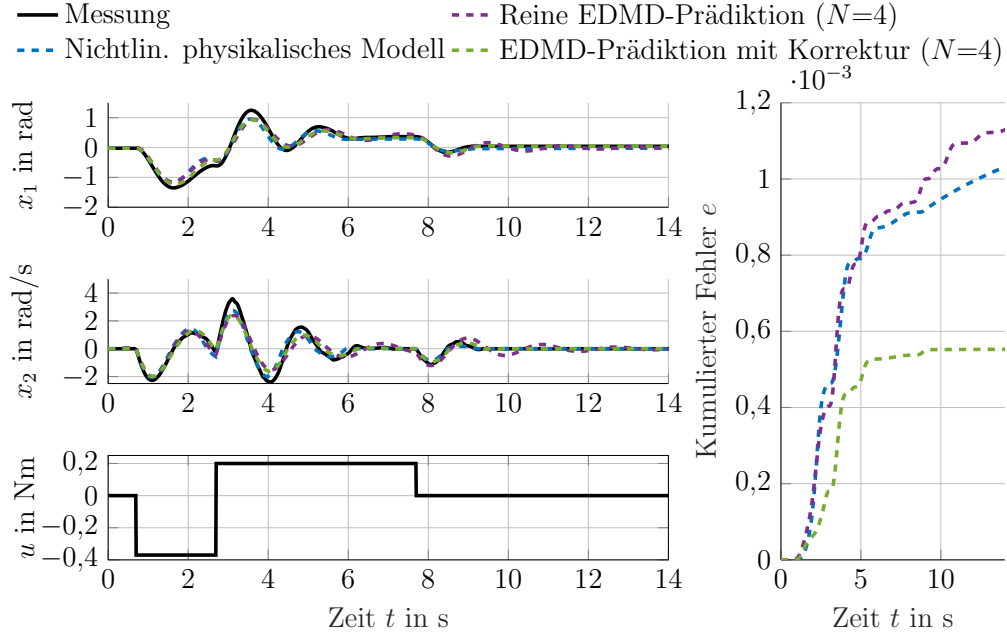


**Abbildung 3.11** Zur Erzeugung der Trainingsdaten wurde der Schlagmechanismus des Golfroboters mit unterschiedlichen Verläufen für  $u$  angeregt und der resultierende Systemausgang  $x_1$  gemessen. Der Verlauf für  $x_2$  wurde offline geschätzt.

Systemverhalten anpassen, vgl. Abbildung 3.13. Dafür wurde zunächst in Anlehnung an [71] ein Algorithmus für ein adaptives EDMD-Modell mit einem rekursiven kleinste-Quadrate-Ansatz realisiert. Durch einen variablen Vergessensfaktor berücksichtigt das Modell vergangene Messdaten in Abhängigkeit der Intensität der auftretenden Systemveränderungen zu vernachlässigen. Die Begrenzung der Spur der Kovarianzmatrix verhindert das einhergehende Risiko eines sogenannten Kovarianz-Windups. Für vollständige Zustandsregelungen, die messbare oder geschätzte Zustandsgrößen erfordern, wurde ein Beobachter integriert, der sich ebenfalls während des Betriebs anpasst. Am Schlagmechanismus des Golfroboters wurde gezeigt, dass das adaptive EDMD-Modell die Regelungsgüte auch bei starken Systemveränderungen deutlich verbessert [70]. Die erfolgreiche Integration der Adaptionsfähigkeit bestätigt die hohe regelungstechnische Verwertbarkeit der EDMD-Modelle gemäß der Kriterien, die im Kapitel 2 formuliert wurden.

## 3.2 Sparse Identification of Nonlinear Dynamical Systems

Motiviert durch Problemstellungen in der Fluidodynamik, entstand in [72] die Methode *Sparse Identification of Nonlinear Dynamics (SINDy)*, welche ein lineares Regressionsmodell basierend auf Messdaten und unter Zuhilfenahme einer nichtlinearen Funktionsbibliothek extrahiert. Damit weist das Verfahren eine nahe Verwandtschaft zu Modellreduktionstechniken wie *Proper Orthogonal Decomposition* und zu Koopman-basierten Techniken wie *Dynamic Mode Decomposition* (vgl. Unterabschnitt 3.1.2) auf. Voraussetzung für SINDy ist die Verfügbarkeit von Mess-



**Abbildung 3.12** Prädiktionsgenauigkeit der EDMD-Verfahren im Vergleich zum nichtlinearen physikalischen Modell am Beispiel des Schlagmechanismus des Golfroboters.

daten des vollständigen Zustands  $\mathbf{x} \in \mathbb{R}^n$ , wobei die Systemordnung  $n$  eine sehr hohe Dimension widerspiegeln kann. Liegen  $M$  Messdaten des Zustands für jeden Zeitschritt  $k = 1, \dots, M$  vor, so können diese zeitversetzt in Matrizen  $\mathbf{X}$  und  $\mathbf{X}'$  nach Gleichung (3.17) angeordnet werden. Um die Evolution des Zustands, welche durch eine autonome Differentialgleichung (3.1) erfolgt, zu modellieren, wird die Dynamik  $\mathbf{f}$  über eine Linearkombination approximiert. Dazu wird eine Funktionsbibliothek aus  $n_\theta$  (nicht-)linearen Termen  $\psi_i(\cdot)$  angenommen, welche die anzunähernde Dynamik  $\mathbf{f}$  basierend auf den Messdaten beschreiben können. Mithilfe der Parametermatrix  $\boldsymbol{\theta} \in \mathbb{R}^{n \times n_\theta}$  kann anschließend der folgende Zusammenhang zwischen den Messdaten ausgedrückt werden:

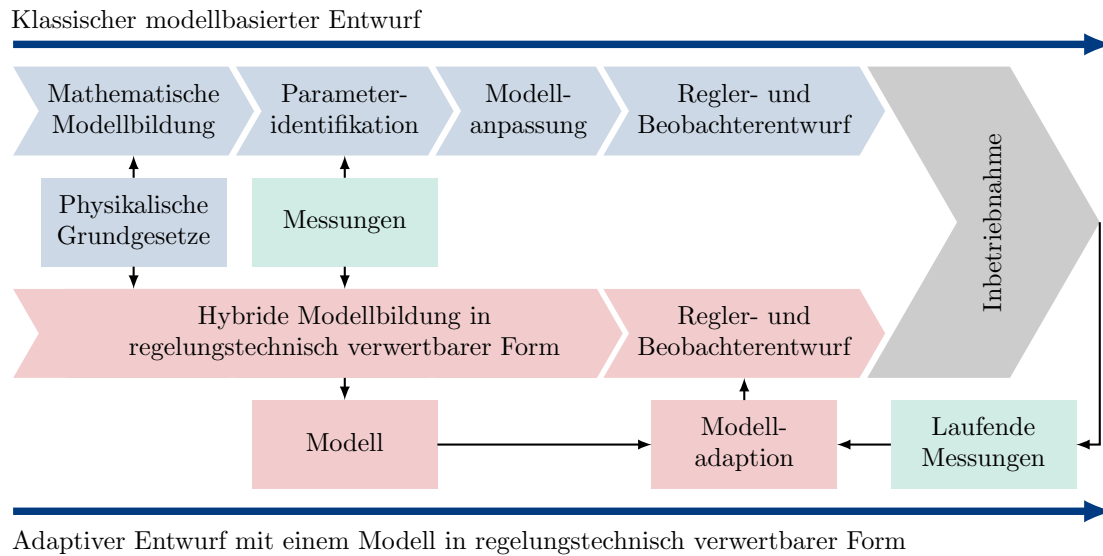
$$\mathbf{X}' = \boldsymbol{\theta} \Psi^T(\mathbf{X}). \quad (3.55)$$

Da jedoch die meisten dynamischen Systeme durch einige wenige Funktionen charakterisiert werden können, wird postuliert, dass die Parametermatrix  $\boldsymbol{\theta}$  dünnbesetzt<sup>14</sup> ist, d. h. dass lediglich einige wenige Einträge verschieden von Null sind. Anschließend lassen sich Algorithmen aus dem Bereich der *Sparse Regression* anwenden. Mittels des populären Ansatzes LASSO<sup>15</sup>, welcher auf [73] zurückgeht und eine Pareto-optimale Lösung bzgl. Modellgüte und Modellkomplexität liefert, lässt sich für die  $i$ -te Zeile von  $\boldsymbol{\theta}$  mit  $i = 1, \dots, n$  die folgende Lösung des Optimierungsproblems formulieren:

$$\boldsymbol{\theta}_i^* = \arg \min_{\boldsymbol{\theta}_i} \|\mathbf{X}'_i - \boldsymbol{\theta}_i \Psi^T(\mathbf{X})\|_2 + \lambda \|\boldsymbol{\theta}_i\|_1. \quad (3.56)$$

<sup>14</sup>im Englischen *sparse*

<sup>15</sup>Least Absolute Shrinkage and Selection Operator



**Abbildung 3.13** Der adaptive Entwurf ermöglicht ein hybrides Modell in regelungstechnisch verwertbarer Form, das sich während des Betriebs anpasst. In Anlehnung an [70].

Hierbei stellt  $\mathbf{X}'_i$  die  $i$ -te Zeile der Datenmatrix  $\mathbf{X}'$  dar. Das konvexe Optimierungsproblem (3.56) lässt sich ebenso auf den Fall nicht-autonomer Systeme erweitern und wird dann als SINDYc bezeichnet<sup>16</sup> [74]. Neben der rein datenbasierten Identifikation nur durch Messdaten kann das SINDy-Verfahren auch Vorwissen berücksichtigen, z. B. durch eine konkrete Bibliothekswahl, bei der Hypothesen über das zu identifizierende System als Kandidatenfunktionen eingehen. Anschließend liegt ein (in der Regel nichtlineares) interpretierbares Modell vor. Allerdings besteht ein gravierender Nachteil darin, dass das SINDy-Verfahren Messwerte des vollständigen Zustands benötigt. Dies ist in realen Prozessen selten umsetzbar aufgrund von messtechnischen Herausforderungen oder des Kostendrucks bzgl. der Sensorik. Dieser Nachteil kann nur relativiert werden, wenn alle Zustände des Systems einen differentiellen Zusammenhang aufweisen, sodass die nicht messbaren Zustandsgrößen alternativ numerisch differenziert oder gefiltert werden können.

#### Beispielhafte Anwendung

Zur Illustration des SINDYc-Verfahrens wird die Dynamik des Golfroboters, welcher im Abschnitt 6.1 vorgestellt wird, anhand von Messdaten identifiziert. Da es sich bei diesem um ein nicht-autonomes System handelt, werden die Stelleingänge nach Gleichung (3.17) ebenfalls in einer Matrix gespeichert und zur Approximation der Dynamik genutzt. Als Messdaten stehen Trajektorien zur Verfügung, die aufgrund einer sinus- (teilweise mit zeitlich variierenden Frequenzen) oder sprungförmigen Anregungen resultieren. Die Aufteilung dieser Messdaten erfolgt dabei wie im Abschnitt 2.2 beschrieben in Trainings-, Validierungs- und Testdaten, um ein konsistentes Lernen des SINDYc-Modells zu erlauben. Nachfolgend ist das Ergebnis einer

<sup>16</sup>Diese Abkürzung bedeutet *SINDy with control* (*SINDYc*). Im weiteren Verlauf wird jedoch der Begriff SINDy genutzt, da sich die Betrachtung autonomer und nicht-autonomer Systeme aus dem Zusammenhang ergibt.

solchen sinusförmigen Testtrajektorie in der Abbildung 3.14 zu sehen, nachdem das SINDYc-Modell mit den Bibliotheken

$$\begin{aligned}\Psi_1(\mathbf{x}, u) &= [1, x_1, x_2, x_1^2, x_2^2, u], \\ \Psi_2(\mathbf{x}, u) &= [1, x_1, x_2, \cos(x_1), \cos(x_2), u], \\ \Psi_3(\mathbf{x}, u) &= [1, x_1, x_2, \cos(x_1), x_2^2, u]\end{aligned}\tag{3.57}$$

trainiert worden ist. Anhand der zeitlichen Verläufe im Vergleich zu den Messdaten ist erkennbar, dass das SINDYc-Modell für jede der verwendeten Bibliotheken eine sehr hohe Modellgüte aufweist, da kaum ein Unterschied zu den Trajektorien der Messung zu sehen ist. Insbesondere unterscheiden sich die Ergebnisse der drei Bibliotheken kaum, da alle einen ähnlich geringen Approximationsfehler liefern und die folgenden Parametermatrizen identifizieren:

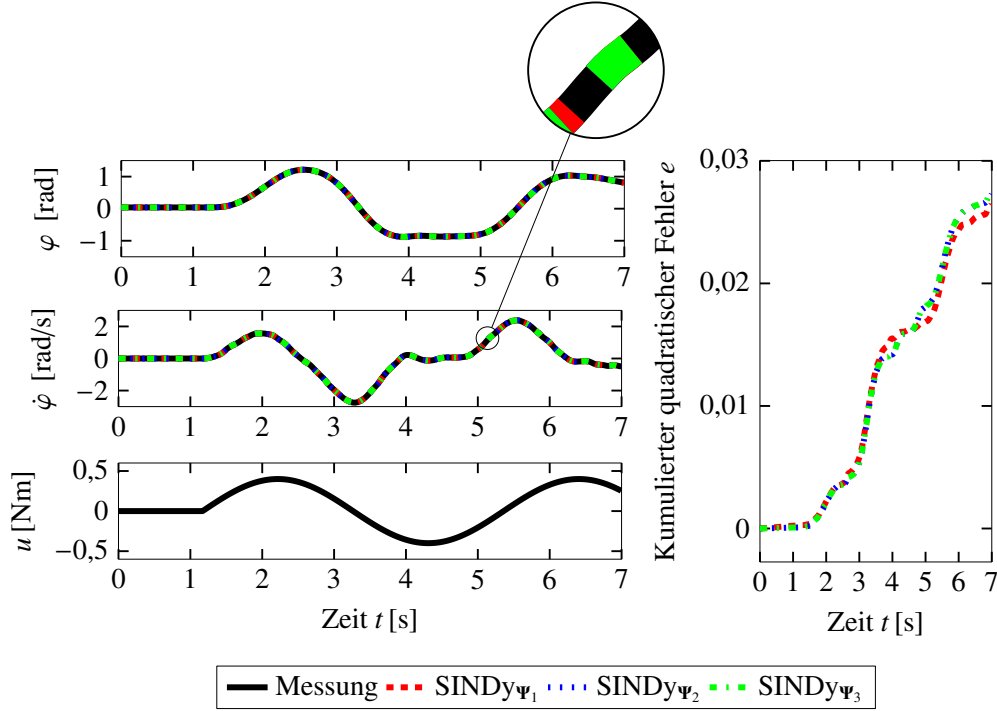
$$\begin{aligned}\boldsymbol{\theta}_1 &= \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & -0,0122 & 0,999 & 0 & 0 & 0,0267 \end{bmatrix}, \\ \boldsymbol{\theta}_2 &= \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 0,0014 & -0,0127 & 0,999 & -0,0012 & 0 & 0,0267 \end{bmatrix}, \\ \boldsymbol{\theta}_3 &= \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 0,0014 & -0,0127 & 0,999 & -0,0012 & 0 & 0,0267 \end{bmatrix}.\end{aligned}\tag{3.58}$$

Das SINDYc-Verfahren ist folglich ein geeignetes Verfahren, um eine physikalisch-interpretierbare Zustandsraumdarstellung zu erhalten, da es bereits das Einbringen von Vorwissen in Form von Bibliotheksfunktionen erlaubt. Allerdings ist es nicht immer gegeben, dass der vollständige Zustand als Messwert vorliegt, sodass diese Methode bzgl. ihrer Grundidee im Kapitel 5 im Rahmen des Beobachterentwurfs erneut aufgegriffen wird.

Das SINDY-Verfahren eignet sich sowohl für autonome als auch nicht-autonome Systeme und besitzt üblicherweise einen geringen Rechenaufwand. Es erfordert Messdaten des **vollständigen** Zustands bzw. seines differentiellen Zusammenhangs sowie geeignet gewählte Bibliotheksterme. Das resultierende, in der Regel nichtlineare Modell ist **interpretierbar** und im weiteren Regelungsentwurf **flexibel** einsetzbar.

### 3.3 Physikalisch motivierte neuronale Netze

Als globale Funktionsapproximatoren können neuronale Netze einen beliebigen funktionalen Zusammenhang abbilden, wenn die Netzarchitektur sowie Trainingsdaten entsprechend gewählt werden [75], [76], [77], [78]. Diese Eigenschaft ist besonders vorteilhaft, um die in Kapitel 1 beschriebene Herausforderung der Modell-Realitäts-Lücke zu adressieren. Stoßen physikalisch motivierte Modelle an ihre Grenzen und sind gleichzeitig Messdaten verfügbar, kann ein neuronales Netz in Kombination mit einem physikalisch motivierten Modell unterstützen und die Modell-Realität-Lücke  $\Delta \mathbf{f}$  ggf. verkleinern [22], [79]. Damit stellen die sogenannten *Physics-Guided Neural*



**Abbildung 3.14** Modellgüte eines SINDYc-Modells unterschieden in die unterschiedlichen Bibliotheken  $\Psi_1, \Psi_2, \Psi_3$  [17] und basierend auf Code aus [28], vgl. [17]

*Networks (PGNNs)*<sup>17</sup> einen sehr populären Ansatz dar, welcher in vielfältigen Disziplinen in unterschiedlichen Strukturen eingesetzt wird [79], [80]. Erste Konstrukte solcher Kombinationen wurden bereits in den 1990er Jahren verwendet, da beispielsweise ein neuronales Netz als Fehlermodell hinter ein physikalisches Modell geschaltet wurde, um den Modellfehler zu lernen [81]. Im Folgenden werden zunächst die Grundlagen neuronaler Netze erläutert, woraufhin physikalisch motivierte Netze definiert und vorgestellt werden.

### 3.3.1 Neuronale Netze

In diesem Abschnitt erfolgt lediglich eine kurze Einführung in die Thematik der neuronalen Netze. Diese soll zum Verständnis der nachfolgenden Methoden beitragen, sodass als weiterführende Literatur [2], [10], [33], [82] empfohlen wird. Die Struktur neuronaler Netze (NN) ist durch die biologische Informationsverarbeitung inspiriert und dient zur Approximation eines beliebigen, funktionalen Zusammenhangs

$$\mathbf{f} : \mathbb{X} \mapsto \mathbb{Y} \quad (3.59)$$

mit den Eingängen  $\mathbf{x} \in \mathbb{X} \subseteq \mathbb{R}^D$  und Ausgängen  $\mathbf{y} \in \mathbb{Y} \subseteq \mathbb{R}^L$ . Dieser Zusammenhang (3.59) wird durch ein neuronales Netz als eine Funktion  $\mathbf{f}_{NN}$  über einen gerichteten Graphen, bestehend aus Knoten und Kanten, beschrieben. Die Knoten, nachfolgend als Neuronen bezeichnet, sind als Verarbeitungseinheiten in mehreren sogenannten versteckten Schichten angeordnet [2], [10], [33], [82]. Die Neuronen sind

<sup>17</sup>Diese sind z. T. auch unter den Begriffen *Physics-Informed Neural Networks (PINNs)* oder *Physics-Based Neural Networks (PBNNs)* bekannt.



hierbei durch gerichtete Kanten verbunden, wobei jeder Kante, die ein Neuron  $i$  mit einem Neuron  $j$  verbindet, ein Gewicht  $\zeta_{ji}$  zugeordnet wird. Somit stellt ein NN mit  $\zeta \in \mathbb{U}$  und  $\hat{\mathbf{y}} \in \mathbb{Y}$  ein nichtlineares Regressionsmodell dar, da es durch

$$\mathbf{f}_{NN} : \mathbb{X} \times \mathbb{U} \mapsto \mathbb{Y}, \quad \hat{\mathbf{y}} = \mathbf{f}_{NN}(\mathbf{x}, \zeta) \quad (3.60)$$

definiert wird. Die Anordnung der Neuronen und ihr Informationsfluss untereinander, der durch die Kantenrichtungen ausgedrückt wird, wird als Netzarchitektur bezeichnet. Es lassen sich grundsätzlich vorwärts gerichtete und rekurrente Architekturen unterscheiden, wobei Letztere einen Informationsfluss der Neuronen durch verschiedene Formen der Rückkopplung des Informationsflusses ermöglicht, z. B. indem der Ausgang eines Neurons mit einem Neuron einer vorhergehenden Schicht verbunden ist. Indem lineare und nichtlineare Aktivierungsfunktionen auf die Neuronen angewendet werden, kann der Zusammenhang Gleichung (3.59) allgemeine nichtlineare Funktionen approximieren. Das Lernen dieses Zusammenhangs erfolgt über das Verfahren *Backpropagation*, welches den prädizierten Ausgang des Netzes  $\hat{\mathbf{y}}_q$  mit den Trainingsdaten  $\mathbf{y}_q$  für  $q = 1, \dots, M$  Messdaten vergleicht [2], [10], [33], [82]. Mithilfe eines Gradientenabstiegsverfahrens wird anschließend die Lösung des Optimierungsproblems

$$\arg \min_{\zeta} J_{err}(\mathbf{x}, \zeta) = \arg \min_{\zeta} \frac{1}{2} \sum_{q=1}^M \|\hat{\mathbf{y}}(\mathbf{x}_q, \zeta) - \mathbf{y}_q\|_2^2 \quad (3.61)$$

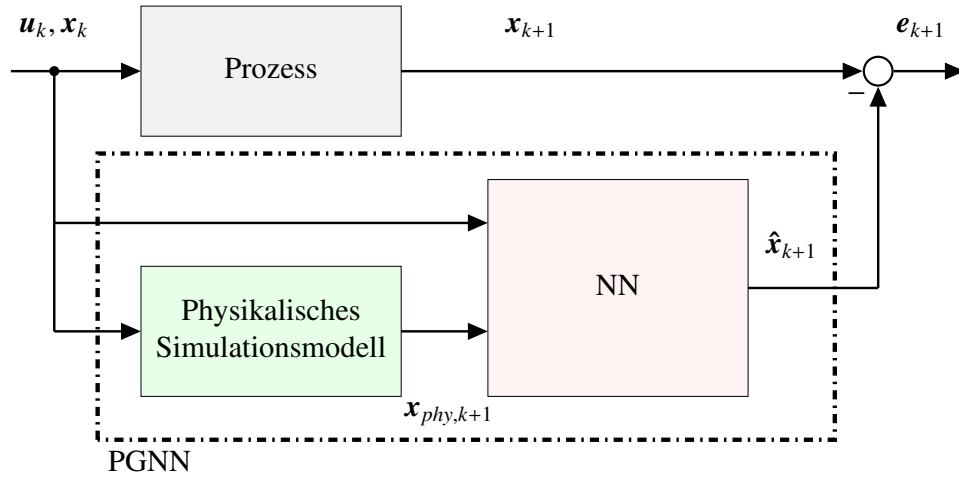
ermittelt. In den Anwendungsbeispielen dieses Kapitels wurde das ADAM-Verfahren genutzt [83]. Durch ihre Struktur stellen NN ein mächtiges Werkzeug in der Funktionsapproximation dar [75], [76], [77], [78] und werden in vielfältigen Situationen eingesetzt. Insbesondere durch die Nutzung vieler versteckter Schichten und verschiedener nichtlinearer Aktivierungsfunktionen ermöglicht ein NN, komplexe Funktionen zu approximieren. Diese tiefen neuronalen Netze, welche viele Schichten aufweisen, werden dem *Deep Learning* zugeordnet, vgl. Abschnitt 2.2. Allerdings besteht beim Lernvorgang neuronaler Netze die Gefahr des Over- oder Underfittings, wenn Einflussgrößen wie Trainingsdaten, Lernrate oder Initialisierung ungünstig gewählt werden [10].

### 3.3.2 Physics-Guided Neural Networks

In den meisten Fällen kann mit geringem Modellierungsaufwand und Vorwissen ein einfaches physikalisches Modell für ein betrachtetes mechatronisches System erarbeitet werden. Für die nachfolgenden Abschnitte wird angenommen, dass ein solches Modell in parametrischer Form vorliegt. Dieses besteht in der Regelungstechnik gewöhnlich aus der folgenden Differentialgleichung mit dem Zustand  $\mathbf{x}_{phy} \in \mathbb{R}^n$  und den Parametern  $\mathbf{p} \in \mathbb{R}^{n_p}$ :

$$\begin{aligned} \dot{\mathbf{x}}_{phy} &= \mathbf{f}_{phy}(\mathbf{x}_{phy}, \mathbf{u}, \mathbf{p}), \\ \mathbf{y}_{phy} &= \mathbf{x}_{phy}. \end{aligned} \quad (3.62)$$

Um die infolge der vereinfachten Modellbildung entstandenen Ungenauigkeiten und Abweichungen des Modells zu kompensieren und eine höhere Modellgüte zu erzielen, wird ein neuronales Netz ergänzt, welches diese approximiert. Voraussetzung für ein überwachtes Lernen der realen Systemdynamik  $\mathbf{f}$  mithilfe neuronaler Netze ist allerdings, dass der vollständige Zustand messbar ist, d. h. dass  $\mathbf{y} = \mathbf{x}$  gilt, oder dieser durch einen differentiellen Zusammenhang ermittelt werden kann (vgl. Abschnitt 3.2). Daher gilt diese Annahme ebenfalls für das physikalische Simulationsmodell (3.62), welches zudem basierend auf einem numerischen Integrationsverfahren diskretisiert wird. Mithilfe der zwei Komponenten, einerseits das physikalisch basierte Simulationsmodell und andererseits ein datengetriebenes neuronales Netz, können je nach Ausprägung der Abweichungen verschiedene Optionen zur Verschaltung der Modellkomponenten in Erwägung gezogen werden [84], beispielsweise eine serielle oder parallele Anordnung. Im Folgenden wird eine Mischform vorgestellt, welche auf [79] zurückgeht. Diese wird als *Physics-Guided Neural Network (PGNN)* bezeichnet und ist in der Abbildung 3.15 durch den gestrichelten Kasten angedeutet. Das PGNN erhält in dieser Struktur als Eingänge nicht nur den prädierten Aus-



**Abbildung 3.15** Struktur eines PGNNs während des Trainings [17]: Es umfasst ein diskretisiertes, physikalisches Simulationsmodell (grün) und ein vorwärts gerichtetes NN (rot).

gang des Simulationsmodells  $\mathbf{x}_{phy,k+1}$ , sondern auch die aktuelle Stellgröße  $\mathbf{u}_k$ , mit welcher das System angeregt wird, und den aktuellen Zustand  $\mathbf{x}_k$ , in dem sich das System befindet. Folglich kann das neuronale Netz nicht nur den temporären Modellfehler lernen, welcher lediglich bei einer Reihenschaltung angenähert wird, sondern auch die vollständige Dynamik sowie die Wechselwirkungen zwischen diesen. Darüber hinaus erleichtert die Berücksichtigung des prädizierten Ausgangs  $\mathbf{x}_{phy,k+1}$  das Training, da es das Netz in eine physikalisch plausible Richtung lenkt, auch wenn das Simulationsmodell (3.62) qualitative Abweichungen aufweist. Der Zusammenhang, den das PGNN beschreibt, kann anschließend durch

$$\mathbf{f}_{PGNN} : \mathbb{R}^n \times \mathbb{R}^n \times \mathbb{R}^p \times \mathbb{U} \mapsto \mathbb{R}^n, \quad \hat{\mathbf{x}} = \mathbf{f}_{PGNN}(\mathbf{x}, \mathbf{x}_{phy}, \mathbf{u}, \zeta) \quad (3.63)$$

ausgedrückt werden [84]. Üblicherweise wird die Dynamik hierbei durch den Vergleich des Datenfehlers (3.61) gelernt. Um allerdings nicht nur ein physikalisch plausibles Modell basierend auf den Trainingsdaten zu entwickeln, sondern auch eine

physikalisch konsistente Prädiktion außerhalb der Trainingsdaten zu gewährleisten, schlagen Karpatne et al. in [79] eine Erweiterung der rein Daten auswertenden Kostenfunktion  $J_{err}$  um einen physikalischen Term  $J_{phy}$  vor. Dieser berücksichtigt qualitatives Vorwissen, welches zusätzlich zu den Daten verfügbar ist. Beispiele dafür sind physikalische Naturgesetze, Erhaltungsprinzipien, Schranken, Regelgesetze oder vages Erfahrungswissen [85], [86]. Dieses Wissen lässt sich durch Gleichheits- oder Ungleichheitsbedingungen  $\mathcal{G}$  bzw.  $\mathcal{H}$  ausdrücken, welche anschließend als weitere Kostenfunktion  $J_{phy}$  formuliert werden können:

$$\begin{aligned}\mathcal{G}(\mathbf{x}) = 0 & \rightarrow J_{phy}(\mathbf{x}, \zeta) = \|\mathcal{G}(\mathbf{x})\|^2, \\ \mathcal{H}(\mathbf{x}) \leq 0 & \rightarrow J_{phy}(\mathbf{x}, \zeta) = \max(0, \mathcal{H}(\mathbf{x})).\end{aligned}\tag{3.64}$$

So erörtern Karpatne et al. beispielsweise die Problematik, wie die Temperatur innerhalb eines Sees abhängig von der Tiefe des Gewässers modelliert werden kann [79]. Mithilfe eines PGNNs, welches ein physikalisch motiviertes Seemodell enthält und die Beziehung zwischen Temperatur, Dichte und Tiefe des Wassers als zusätzliche Wissenskomponente  $J_{phy}$  über eine Bedingung  $\mathcal{H}$  im Lernprozess berücksichtigt, kann schließlich ein Modell mit hoher Güte entwickelt werden. Entgegen der Verwendung der physikalischen Kostenfunktion als Regularisierungsterm in [79], wird in der Dissertation [17] ein Mehrzieloptimierungsproblem formuliert, sodass

$$\zeta^* = \arg \min_{\zeta} J(\mathbf{x}, \zeta) = \arg \min_{\zeta} (1 - \lambda_{phy}) \cdot J_{err}(\mathbf{x}, \zeta) + \lambda_{phy} \cdot J_{phy}(\mathbf{x}, \zeta).\tag{3.65}$$

gilt. Dieses Mehrzieloptimierungsproblem (3.65), welches während des Trainings gelöst werden soll, wird mit einem einfachen Skalarisierungsverfahren umgesetzt [87], [88]. Die gewichtete Summe legt durch den Skalar  $0 \leq \lambda_{phy} \leq 1$  die Priorisierung der einzelnen Ziele  $J_{\bullet}$  fest. Das Auffinden der pareto-optimalen Punkte gestaltet sich in der Praxis aber herausfordernd. Daher diskutieren die Autoren in [89] verschiedene Einflussfaktoren auf die Konvergenz des Mehrzieloptimierungsproblems (3.65), wenn im Gegensatz zum vorgestellten Konzept die Differentialgleichung des Simulationsmodells (3.62) als physikalische Kostenfunktion  $J_{phy}$  betrachtet wird, um das Netz direkt die Systemdynamik lernen zu lassen. Diese populäre Strategie [90], [91], [92], [93], [94], das physikalische Simulationsmodell statt qualitativem Vorwissen in den Lernprozess zu integrieren, ist jedoch nur zielführend, wenn das Simulationsmodell eine hohe Modellgüte aufweist, da das Netz potentiell irrtümliche Eigenschaften nachbilden kann. Am Ausgangspunkt in diesem Beitrag steht allerdings ein Simulationsmodell, welches größere Abweichungen zur Systemdynamik aufweist, sodass diese Strategie nicht in Frage kommt (vgl. Kapitel 1). Dennoch zeigt [89] auf, dass insbesondere jene Parameter, welche die Struktur des Netzes bestimmen, einen starken Einfluss auf die Form der Paretomenge besitzen. Empfehlenswert ist daher eine sorgsame Wahl dieser mittels der Hyperparameteroptimierung. Darüber hinaus steigt die Komplexität des Optimierungsproblems, wenn eine Anpassung für mehr als zwei Ziele in der Kostenfunktion (3.65) vorgenommen wird [87], [88]. In [17] sowie in [84] wurde das Konstrukt des PGNNs erstmals auf ein nicht-autonomes System aus der Mechatronik angewendet. In dieser Disziplin existiert üblicherweise qualitatives Vorwissen, welches in der Kostenfunktion durch einen Mehrzieloptimierungsansatz berücksichtigt werden kann, z. B. die Energiebilanz eines Systems.

### Beispielhafte Anwendungen

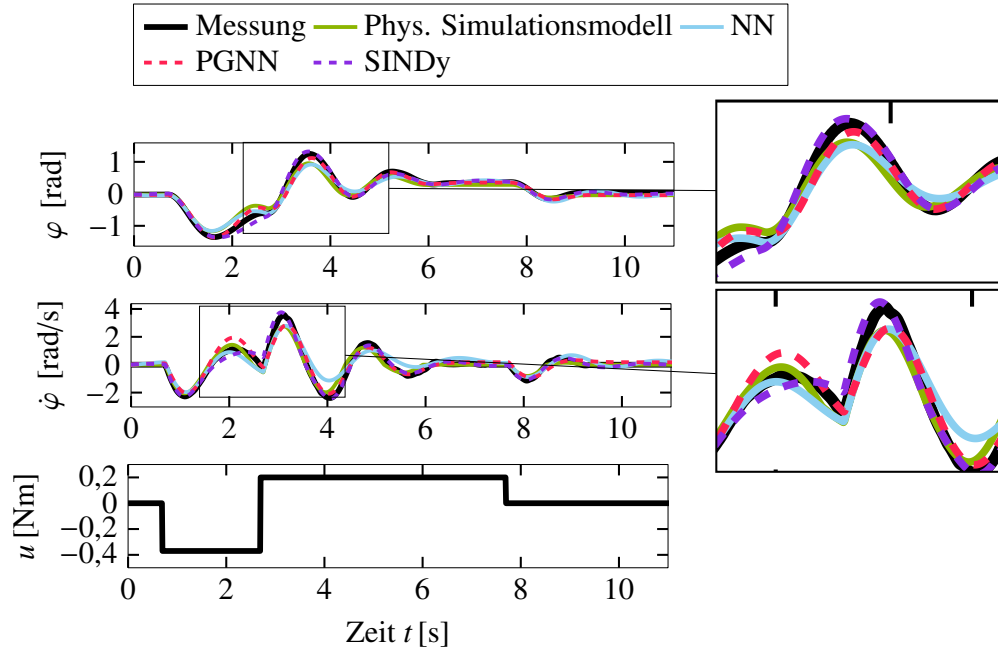
**Golfroboter** Der Golfroboter, welcher in Abschnitt 6.1 vorgestellt wird und bereits in vorangegangenen Abschnitten als Testobjekt genutzt wurde (vgl. Absatz 3.1.4 und Abschnitt 3.2), besitzt ein komplexes, nichtlineares Reibmoment  $M_d$ , vgl. Gleichung (6.11). Dieses approximiert die Mischung aus Haft- und Gleitreibung, weist aber dennoch in manchen Situationen Diskrepanzen zum tatsächlichen Verhalten des Golfroboters auf. Daher soll basierend auf dem nichtlinearen Modell (6.11) ein PGNN entwickelt werden, dessen Modellgüte nach einem Training mit rein datenbasierten als auch physikalisch motivierten Modellen verglichen werden soll. Zusätzliches Wissen wird während des Trainings durch die Energiebilanz des Golfroboters

$$\begin{aligned} J_{phy}(\ddot{\varphi}, \dot{\varphi}, \varphi, u) &= \Delta E(\ddot{\varphi}, \dot{\varphi}, \varphi, u) \\ &= J\ddot{\varphi}\dot{\varphi} + mga\dot{\varphi}\sin(\varphi) + d\dot{\varphi}^2 + r\mu\text{sign}(\dot{\varphi})|\dot{\varphi}|^2a + mg\cos(\varphi)|\dot{\varphi} - 4u\dot{\varphi} \end{aligned} \quad (3.66)$$

mittels der Gleichheitsnebenbedingung aus Gleichung (3.64) berücksichtigt. Anschließend werden verschiedene Trajektorien infolge unterschiedlicher Anregungen zum Lernen genutzt, indem diese Daten für das Training, die Validierung und zum Testen aufgeteilt werden [17], [95]. Ferner wird eine Hyperparameteroptimierung mithilfe einer Bayesschen Optimierung durchgeführt (vgl. Unterabschnitt 4.4.3, um beispielsweise die Anzahl der Neuronen oder die Gewichtung  $\lambda_{phy}$  zu bestimmen. Nachdem das PGNN erfolgreich mit der optimierten Parametrierung  $\lambda_{phy} = 0,8175$  und einer Anzahl von 27 Neuronen trainiert wurde, wird es mit einer unbekannten Trajektorie getestet sowie mit weiteren Modellen verglichen<sup>18</sup>. Schließlich zeigt Abbildung 3.16 Ergebnisse dieses Vergleichs, indem die Trajektorien der Modelle bzgl. der schwarz gekennzeichneten Testtrajektorie dargestellt werden. Es ist erkennbar, dass alle Modelle zunächst eine zufriedenstellende Modellgüte aufweisen. Werden jedoch Abschnitte der Abbildung vergrößert, zeigen sich qualitative Unterschiede. So schwankt das in rot dargestellte PGNN meist zwischen dem Verlauf der Messdaten und des physikalischen Modells (in grün), kann aber in der Regel eine Verbesserung der Modellgüte im Vergleich zum physikalischen Modell erzielen, z. B. für den Winkel. Dagegen weicht das NN (in hellblau) am stärksten vom tatsächlichen Verlauf des Golfroboters ab. Folglich scheint das PGNN die Vorteile beider Perspektiven zu vereinen und daraus ein qualitativ höherwertiges Modell zu erschaffen. Allerdings übertrifft das SINDy-Modell, dessen Trajektorien in lila dargestellt sind, die Approximationsgüte des PGNN-Modells deutlich.

**Servoventil** Servoventile bilden als Teil der Aktorik eine zentrale Rolle in vielen Applikationen. Jedoch ist die Modellierung ihrer Dynamik aufgrund starker Nichtlinearitäten herausfordernd und aufwendig [96]. Häufig wird je nach Einsatzzweck eine erforderliche Modellierungstiefe gewählt, um Ressourcen effizient zu nutzen [97]. Im Kontrast zu einem detaillierten nichtlinearen Modell wie in [96] ist beispielsweise das folgende Modell, welches eine Verzögerungsdynamik zweiter Ordnung aufweist, in der Lage, die grundlegende Systemdynamik des Ventils abzubilden und stellt folglich

<sup>18</sup>Das SINDy-Modell basiert auf der Bibliothek  $\Psi(\mathbf{x}, u) = [x_1, x_2, \sin(x_1), \cos(x_2), \tan(x_2)^{-1}, u]^T$ .



**Abbildung 3.16** Modellgüte eines PGNNs im Vergleich zu anderen datengetriebenen und physikalisch basierten Modellen anhand des Golfroboters, vgl. [17], [84]

einen Kompromiss bzgl. der Modellierungstiefe und -genauigkeit dar:

$$\begin{aligned} \dot{\mathbf{x}} &= \begin{bmatrix} x_2 \\ -\frac{2D}{T}x_2 - \frac{1}{T^2}x_1 + \frac{K}{T^2}u \end{bmatrix}, \\ y &= x_1. \end{aligned} \quad (3.67)$$

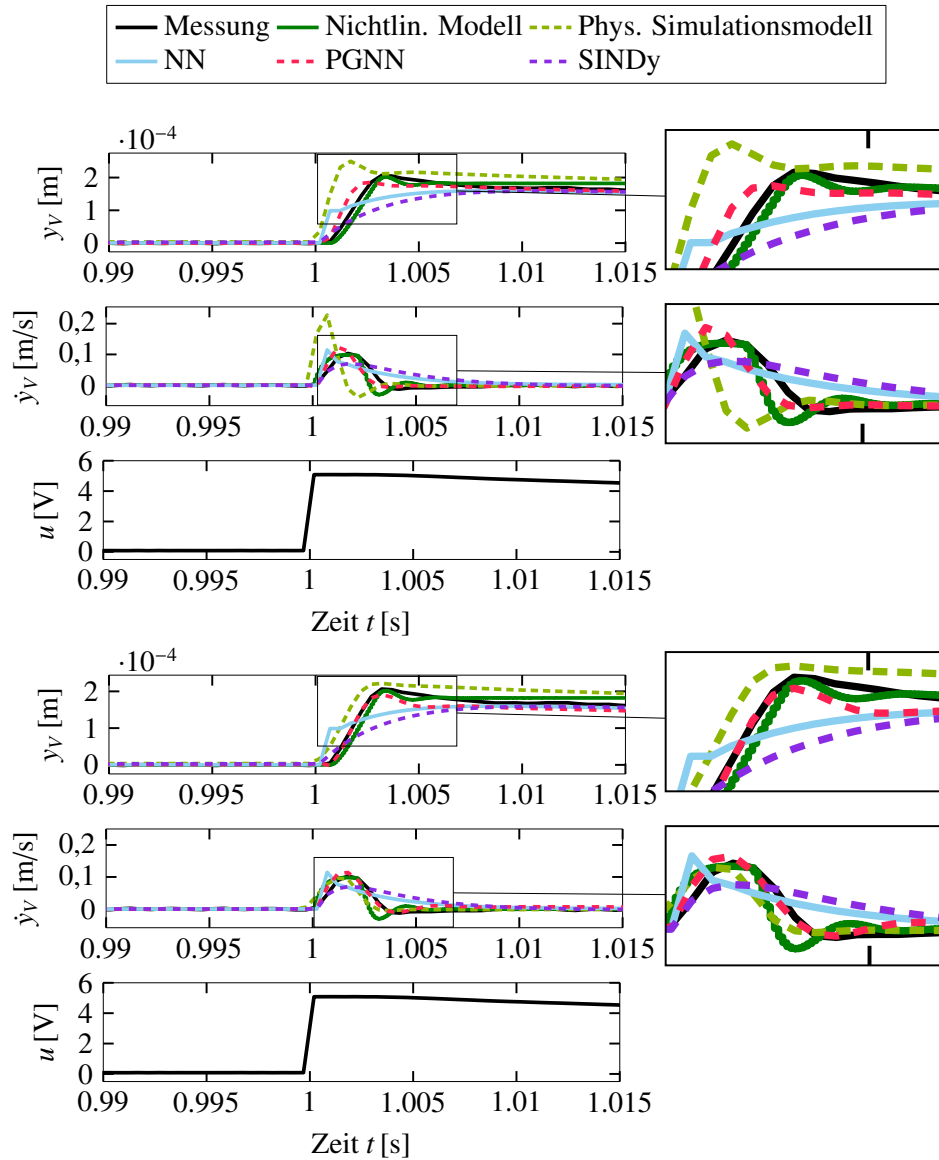
Der Systemzustand  $\mathbf{x} = [y_V, \dot{y}_V]^T$  beinhaltet die Ventilschieberposition  $y_V$  sowie dessen Geschwindigkeit  $\dot{y}_V$ , während das Ventil als Eingang die Spannung  $u$  erhält. Die Parameter des Modells Gleichung (3.67) lauten  $\mathbf{p} = [K, D, T]^T = [0, 1, 0, 5, 1/350]^T$ . Aufgrund der Viskosität des Öls existieren allerdings Beschränkungen der Ventilschiebergeschwindigkeit und -beschleunigung, die das Modell (3.67) nicht wiedergeben kann. Durch mehrere Versuche können diese Beschränkungen experimentell angenähert und in das Modell (3.67) integriert werden, welches folglich eine tieferen Modellierungsgrad erhält. Im Folgenden wird anhand der Abbildung 3.17 die Auswirkung des Modellierungsgrads im Kontext der PGNN deutlich, indem das Modell (3.67) jeweils mit und ohne empirische Beschränkungen als physikalisches Simulationsmodell genutzt wird. Das Training umfasst wie im vorigen Beispiel die Verwendung einer Energiebilanz als zusätzliches Vorwissen und nutzt Trainingsdaten, welche aus verschiedenen Sprunganregungen resultieren (vgl. [17], [84]). Die Visualisierungen in Abbildung 3.17 unterscheiden sich somit nur durch die Verläufe des PGNNs und des physikalischen Simulationsmodells. Beide PGNNs weisen nach der Hyperparameteroptimierung 11 Neuronen auf, unterscheiden sich jedoch in der Gewichtung des physikalischen Kostenterms zu  $\lambda_{phy} = 0,2527$  (Modell ohne Beschränkungen) und  $\lambda_{phy} = 0,3206$  (Modell mit Beschränkungen). Erneut wird die Modellgüte der beiden PGNNs mit anderen Modellen verglichen, u. a. mit der eines SINDy-Modells, welches aus einem Training basierend auf einer Bibliothek mit

Zuständen und Eingängen resultiert, um die Vergleichbarkeit zum Modell (3.67) zu gewährleisten. In der oberen Darstellung der Abbildung 3.17 weist das physikalische Modell durch die fehlenden Beschränkungen große Abweichungen zu den Messdaten auf, welche sich insbesondere in der Geschwindigkeit des Ventilschiebers durch ein Überspringen bemerkbar machen. Trotzdem gelingt es dem PGNN, eine sehr gute Approximation des Ventils zu erzielen, welche auf dem Niveau des nichtlinearen Ventilmodells liegt. Dies lässt sich beispielsweise in den beiden Vergrößerungen rechts von der Abbildung erkennen. Im Kontrast dazu schaffen es weder das NN noch das SINDy-Modell das Systemverhalten anzunähern und beide weisen starke qualitative Abweichungen auf. Dies resultiert aus der einfach gewählten Bibliothek des SINDy-Modells. Erhielte dieses eine Bibliothek mit Elementen des nichtlinearen Modells, wäre eine höhere Modellgüte zu erwarten. Wird für das PGNN ein Modell mit höherem Detaillierungsgrad verwendet, wie z. B. durch Berücksichtigung der Beschränkungen, verbessert sich die Modellgüte enorm. Die untere Visualisierung der Abbildung 3.17 verdeutlicht dieses Erkenntnis, da die Vergrößerungen eine genauere Approximation als das nichtlineare Modell sowohl für die Ventilschieberposition als auch für die -geschwindigkeit zeigen. Dieses Anwendungsbeispiel zeigt folglich auf, dass die Modellgüte eines PGNNs u. a. explizit von der Qualität des verwendeten physikalischen Simulationsmodells abhängig ist. Die Nutzung des PGNNs erlaubt deshalb eine Ersparnis des Modellierungsaufwands, da ein nichtlineares Ventilmodell aus einer zeitintensiven und personell aufwendigen Entwicklung hervorgeht. Dieser Entwicklungszyklus kann durch das Training und die Nutzung eines PGNNs bei Gewährleistung einer vergleichbar hohen Modellgüte deutlich reduziert werden.

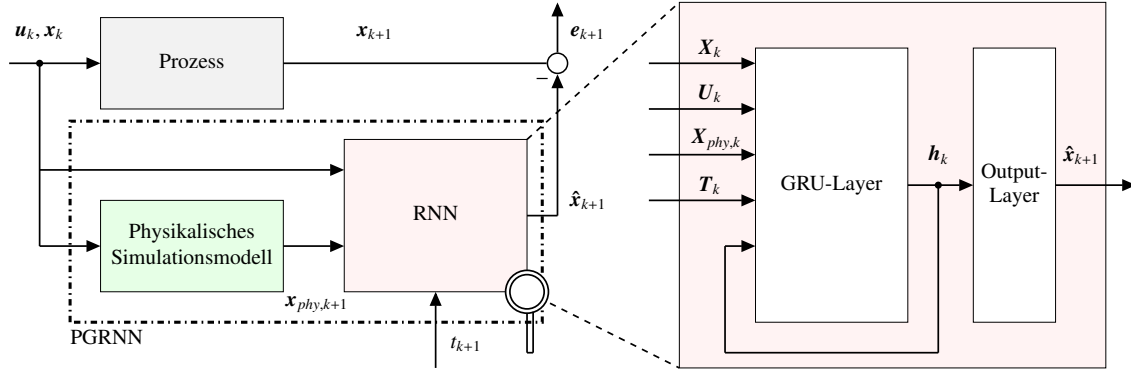
### 3.3.3 Physics-Guided Recurrent Neural Networks

Für Zeitreihendaten wird in der Regel eine rekurrente Architektur empfohlen, um den sequentiellen Verlauf und dessen Langzeiteffekte abbilden zu können [2], [77], [82]. Daher wird das in Unterabschnitt 3.3.2 entwickelte PGNN durch die Nutzung eines rekurrenten Netzes angepasst. Diese adaptierte Struktur wird als *Physics-Guided Recurrent Neural Network (PGRNN)* bezeichnet und beispielsweise für das bereits erwähnte Beispiel zur Schätzung von Temperaturen in Flüssen und Gewässern eingesetzt [98]. Das PGRNN  $f_{PGRNN}$ , welches in der Abbildung 3.18 dargestellt ist, weist die gleiche Grundstruktur aus physikalischem Simulationsmodell und datenbasiertem Anteil wie das PGNN in der Abbildung 3.15 auf. Es unterscheidet sich dementsprechend nur durch den zusätzlichen Eingang der Zeit  $t \in \mathbb{R}$  sowie durch die innere Architektur des neuronalen Netzes. Dieses ist in der Abbildung 3.18 vergrößert dargestellt und besitzt analog zur Modellierung in der Regelungstechnik einen intrinsischen Zustand, den *Hidden State*  $\mathbf{h}_k \in \mathbb{R}^Z$  mit  $Z$  Neuronen, der zeitlich aufeinander folgende Daten durch eine Rückführung berücksichtigt.

Basierend auf bestehenden Formulierungen des PGRNNs (vgl. [98]) wird dieses Konstrukt in der Dissertation [17] sowie in [95] für nicht-autonome, mechatronische Systeme erweitert. Daher hängt  $\mathbf{h}_k$  sowohl von dem vorherigen Zustand  $\mathbf{h}_{k-1}$  als auch von den aktuellen Eingängen ab. Diese umfassen jeweils  $M$  vergangene, zeitliche



**Abbildung 3.17** Modellgüte eines PGNNs bestehend aus einem Simulationsmodell mit unterschiedlichen Modellierungstiefen im Vergleich zu anderen datengetriebenen und physikalisch basierten Modellen anhand der Ventildynamik: Simulationsmodell (3.67) ohne Beschränkungen (oben), Simulationsmodell (3.67) mit Beschränkungen (unten), vgl. [17], [84].



**Abbildung 3.18** Struktur eines PGRNNs während des Trainings [17] (vgl. [95]): Es umfasst ein physikalisches Simulationsmodell (grün) und ein RNN (rot), welches ein GRU-Layer gefolgt von einer Ausgabeschicht enthält.

Verläufe bis zum aktuellen Zeitpunkt  $k$ , sodass Folgendes gilt:

$$\begin{aligned} \mathbf{X}_k &:= (\mathbf{x}_0, \mathbf{x}_1, \dots, \mathbf{x}_k), \\ \mathbf{U}_k &:= (\mathbf{u}_0, \mathbf{u}_1, \dots, \mathbf{u}_k), \\ \mathbf{X}_{phy,k} &:= (\mathbf{x}_{phy,1}, \mathbf{x}_{phy,2}, \dots, \mathbf{x}_{phy,k+1}), \\ \mathbf{T}_k &:= (t_0, t_1, \dots, t_k). \end{aligned} \quad (3.68)$$

Die Verarbeitung der sequenziellen Daten kann allerdings zu numerischen Herausforderungen führen, welche mit der Einführung der *Long Short-Term Memory (LSTM)*-Zelle [99] und der Weiterentwicklung dieser zur *Gated Recurrent Unit (GRU)*-Zelle behoben werden konnten [100]. Deshalb nutzt das PGRNN die GRU-Zelle, welche die Daten durch *Update Gates* und *Reset Gates* verarbeitet. Das Update Gate  $\mathbf{z}_k$  übergibt dem Hidden State neue Informationen, während das Reset Gate  $\mathbf{r}_k$  sukzessiv andere, nicht mehr relevante Informationen löscht. Es resultieren die folgenden Gleichungen für den Eingang  $\mathbf{s}_k = [\mathbf{x}_k, \mathbf{u}_k, \mathbf{x}_{phy,k}]^T$ , welche zur Aktualisierung des Zustands  $\mathbf{h}_k$  durch eine Linearkombination des vorherigen Zustands  $\mathbf{h}_{k-1}$  und des Zustands  $\tilde{\mathbf{h}}_k$  genutzt werden (vgl. [17], [95]):

$$\begin{aligned} \mathbf{z}_k &= \sigma_g(\mathbf{W}_z \mathbf{s}_k + \mathbf{b}_z + \mathbf{R}_z \mathbf{h}_{k-1}), \\ \mathbf{r}_k &= \sigma_g(\mathbf{W}_r \mathbf{s}_k + \mathbf{b}_r + \mathbf{R}_r \mathbf{h}_{k-1}), \\ \tilde{\mathbf{h}}_k &= \sigma_s(\mathbf{W}_{\tilde{h}} \mathbf{s}_k + \mathbf{b}_{\tilde{h}} + \mathbf{r}_k \odot (\mathbf{R}_{\tilde{h}} \mathbf{h}_{k-1})), \\ \mathbf{h}_k &= (1 - \mathbf{z}_k) \odot \mathbf{h}_{k-1} + \mathbf{z}_k \odot \tilde{\mathbf{h}}_k. \end{aligned} \quad (3.69)$$

Die Gewichte sind nach Eingangsgewichten  $\mathbf{W}_\bullet$ , rekurrenten Gewichten  $\mathbf{R}_\bullet$  und additiven Gewichten  $\mathbf{b}_\bullet$  aufgeteilt, wohingegen die Aktivierungsfunktion der Gates  $\sigma_g$  jeweils durch eine logistische Funktion und die Aktivierungsfunktion des Hidden States  $\sigma_s$  durch eine Tangens-Hyperbolicus-Funktion abgebildet werden. Anschließend folgt wie in der Abbildung 3.18 zu erkennen eine Ausgabeschicht, welche aus dem hochdimensionalen Zustand  $\mathbf{h}_k$  den tatsächlichen Systemzustand  $\hat{\mathbf{x}}_{k+1}$  extrahiert.

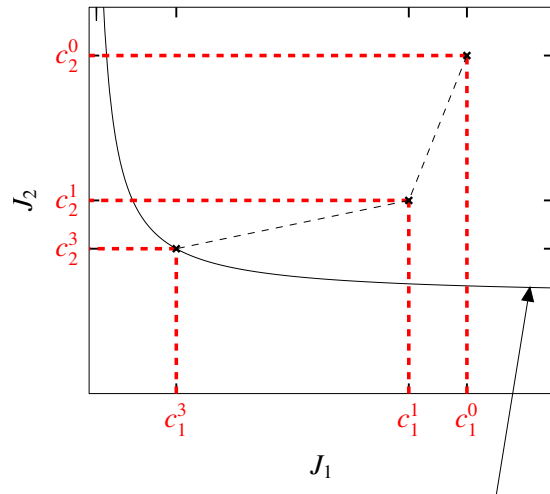
In Analogie zum PGNN wird für das PGRNN ebenfalls physikalisch konsistentes Lernen während des Trainings angestrebt. Obwohl die gewichtete Summe (3.65)



meist die erste Wahl in vielen Publikationen darstellt [89], weist diese verschiedene Nachteile auf [87], [88]. Daher wird eine komplexere Strategie zur Lösung des Mehrzieloptimierungsproblems umgesetzt: Die Gütevektoroptimierung nach Kreißelmeier und Steinhauser [24], [101]. Grundlegende Idee des Verfahrens stellt die Konstruktion von Grenzen für jede der einzelnen Kostenfunktionen  $J_i$  mit  $i = 1, \dots, N_J$  dar, die diese jeweils sukzessiv verkleinern. Zu Beginn jeder Lerniteration  $\tilde{i} = 0, \dots, \varrho$  werden diese Grenzen  $c_i^{\tilde{i}}$  mit  $c_i^{\tilde{i}} > J_i^{\tilde{i}}(\cdot)$  initialisiert, woraufhin das Maximum der  $N_J$  genormten Kostenfunktionen gesucht wird:

$$J(\cdot) = \max \left\{ \frac{J_1^{\varrho}(\cdot)}{c_1^{\varrho}}, \dots, \frac{J_{N_J}^{\varrho}(\cdot)}{c_{N_J}^{\varrho}} \right\}, \quad \text{mit} \quad J_i^{\varrho}(\cdot) \leq c_i^{\varrho} \leq c_i^{\varrho-1} \leq \dots \leq c_i^0. \quad (3.70)$$

Dadurch können alle Gütemaße  $J_i$  schrittweise verkleinert werden, wie es exemplarisch in der Abbildung 3.19 für  $N_J = 2$  Kostenfunktionen und  $\varrho = 3$  Iterationen dargestellt ist. Dieses Vorgehen wird durchgeführt, bis entweder eine maximale Anzahl an Iterationen erreicht worden ist oder keine Verringerung der Grenze  $c_i$  mehr erreicht werden kann. Dies geschieht beispielsweise, wenn der Wert einer Kostenfunktion  $J_i$  bereits sehr niedrig ist oder ein Pareto-optimaler Punkt gefunden worden ist (vgl. Abbildung 3.19). Ein PGRNN, welches mit dieser Mehrzieloptimierungsstrategie während des Trainings betrieben worden ist, wird als *Multi-Objective Physics-Guided Recurrent Neural Network (MOPGRNN)* bezeichnet (vgl. [17], [95]).

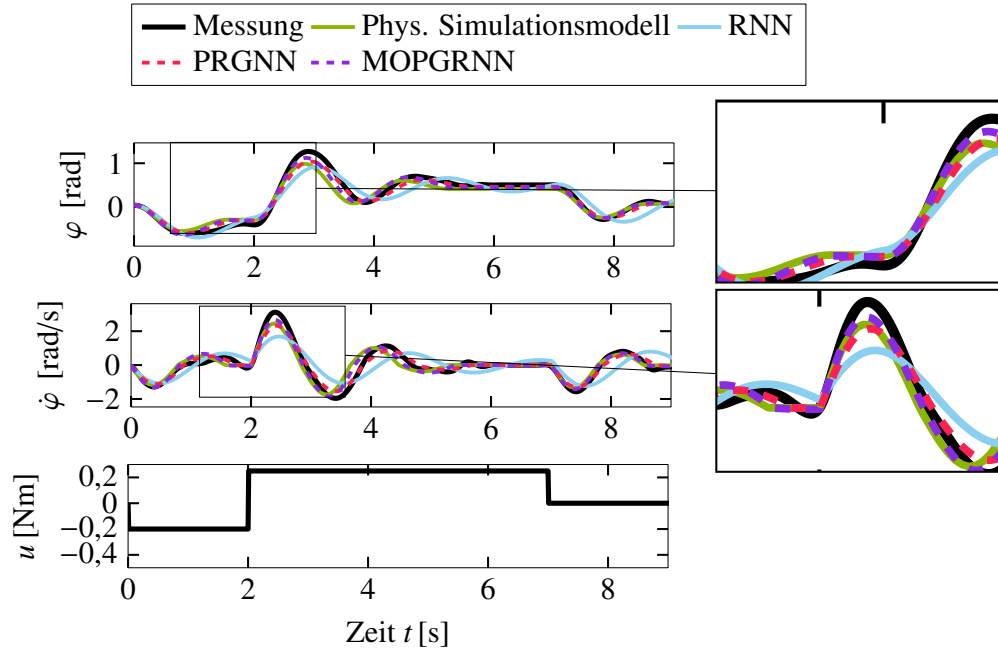


Menge der Pareto optimalen Punkte

**Abbildung 3.19** Schematischer Ablauf einer Gütevektoroptimierung für zwei konkurrierende Ziele  $J_1$  und  $J_2$  und drei Iterationen [17]

### Beispielhafte Anwendung

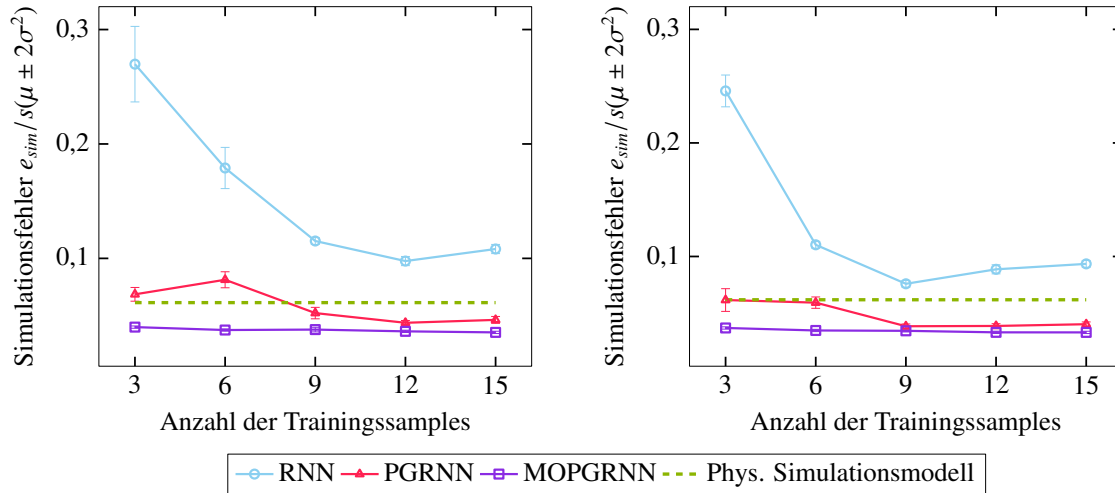
**Golfroboter** Analog zum Anwendungsbeispiel des PGNNs wird die Modellgüte eines PGRNNs bzw. MOPGRNNs anhand des Golfroboters analysiert. Hierbei unterscheiden sich PGRNN und MOPGRNN lediglich in der gewählten Strategie zur Lösung des Mehrzieloptimierungsproblems (3.66). Es werden weiterhin das physi-



**Abbildung 3.20** Modellgüte eines PGRNNs bzw. MOPGRNNs im Vergleich zu anderen datengetriebenen und physikalisch basierten Modellen anhand des Golfroboters, vgl. [17], [95]

kalische Simulationsmodell (6.11) sowie die Energiebilanz (3.66) verwendet. Ebenso stehen dieselben Messdaten des Golfroboters bereit, um das PGRNN und das MOPGRNN zu trainieren. Allerdings werden die Messdaten in verschiedene Datensätze mit unterschiedlichem Umfang aufgeteilt, um die Abhängigkeit der Modellgüte von der Datenmenge zu untersuchen. Folglich kann ein Training entweder basierend auf wenigen Datensätzen (drei bis sechs Trainingssamples) oder vielen Datensätzen (12-15 Trainingssamples) durchgeführt werden. Exemplarisch ergibt sich nach der Optimierung der Hyperparameter und einem Training mit sechs Samples das Modellverhalten, welches in der Abbildung 3.20 zu sehen ist. Die Modellgüte des PGRNNs und MOPGRNNs wird hierbei jeweils mit den Messdaten des Golfroboters, dargestellt durch die schwarzen Trajektorien, und ihren einzelnen Modellkomponenten, dem physikalischen Simulationsmodell (in grün) und dem rekurrenten neuronalen Netz (in blau), verglichen. Im Vergleich aller Modelle ist deutlich erkennbar, dass das rekurrente neuronale Netz (RNN) die geringste Modellgüte aufweist, da es qualitativ das Verhalten des Golfroboters abbilden kann, aber große Abweichungen bestehen. Dies zeigen insbesondere die Vergrößerungen auf der rechten Seite der Abbildung. Das PGRNN, welches ohne die Gütevektroptimierung trainiert worden ist, erzielt meist eine Verbesserung der Modellgüte im Vergleich zum physikalischen Simulationsmodell, beispielsweise in der Approximation der Winkelgeschwindigkeit. Wird nun eine komplexere Strategie statt eines einfachen Skalarisierungsverfahrens zur Lösung des Mehrzieloptimierungsproblems umgesetzt, verbessert sich die Modellgüte des PGRNNs erneut: Die Trajektorien des MOPGRNNs, dargestellt in violett, geben die Dynamik des Golfroboters am besten wieder.

Wegen des Einflusses der initialisierten Gewichte auf die Performanz eines neuronalen Netzes, wird eine statistische Untersuchung vorgenommen, um die Datenabhängigkeit zu ermitteln. Dazu wird jede der drei Netzarten 16 Mal mit denselben



**Abbildung 3.21** Statistische Auswertung des Simulationsfehlers verschiedener Modellarten für den Golfroboter, unterschieden nach der Anzahl der Trainingssamples und danach, ob die Trainingsdaten sinusförmige Anregungen enthielten (rechte Grafik) oder nicht (linke Grafik) [17], [95]

Einstellungen trainiert. Für jedes dieser Netze wird daraufhin die Modellgüte mittels derselben Testtrajektorie evaluiert und anhand eines Simulationsfehlers  $e_{sim}$  ausgewertet, der die Fläche zwischen der tatsächlichen und der prädizierten Trajektorie bestimmt [17], [95]. Aufgrund der statistischen Auswertung der verschiedenen Versuche lassen sich der durchschnittliche Fehler  $\mu$  und die Standardabweichung  $\sigma$  berechnen, welche in der Abbildung 3.21 durch einen Punkt bzw. eine vertikale Linie dargestellt sind. Ferner zeigt die Abbildung die Datenabhängigkeit, indem die  $x$ -Achse die Anzahl der genutzten Trainingssamples beschreibt [17], [95]. Des Weiteren wird in ein Training mit (rechte Grafik) und ohne (linke Grafik) sinusförmigen Anregungen unterschieden, da der Golfroboter insbesondere bei sinusförmigen Anregungen seine starken nichtlinearen Effekte aufzeigt (vgl. Abschnitt 6.1). Unabhängig von den verwendeten Daten und der Anzahl der Trainingssamples ist auffällig, dass das RNN grundsätzlich den anderen Modellen unterliegt und nicht an die Modellgüte des physikalischen bzw. hybriden Modells reichen kann. Dagegen zeigt das MOPGRNN eine Robustheit bzgl. der verwendeten Daten auf, da dessen Modellgüte unabhängig von der Datenmenge und der Anzahl der verwendeten Samples gleichbleibend hoch ist. Diese ist sogar höher als die Modellgüte des physikalischen Modells, wie der Simulationsfehler aufzeigt. Folglich verbessert die Mehrzieloptimierung mittels der Gütevektoroptimierung nicht nur die Modellgüte des MOPGRNNs, sondern erhöht zudem die Robustheit des Netzes im Vergleich zum PGRNN, welches mittels einer gewichteten Summe trainiert wird und eine Datenabhängigkeit aufweist.

Physikalisch motivierte neuronale Netze erfordern ein einfach auszuwertendes, physikalisches Simulationsmodell sowie ein u. U. aufwendiges, architekturabhängiges Training. Vorteile stellen die Berücksichtigung von zusätzlichem physikalischem Vorwissen dar sowie die **physikalische Plausibilität** des resultierenden Modells. Dieses weist in der Regel eine sehr **hohe Modellgüte** auf, setzt aber Messdaten des **vollständigen** Zustands voraus.

## 3.4 Datengetriebene PCHD-Modelle

Motiviert durch den Begriff der regelungstechnischen Verwertbarkeit von Streckenmodellen, vgl. [16], und inspiriert durch die herausragenden Vorteile der Passivitätseigenschaft wurde in [102] erstmals eine neue Methode zur datengetriebenen Bestimmung von PCHD-Modellen vorgeschlagen. Nach einer kurzen Einführung des Passivitätskonzepts sowie der PCHD-Modelle wird die Methode vorgestellt und anhand von Beispielen illustriert.

### 3.4.1 PCHD-Modelle

Der Begriff der Passivität adressiert Dissipationseffekte dynamischer Systeme [103], [104]. Eng damit verbunden ist der Begriff der Hyperstabilität, der sich auf lineare Systeme mit einer positiv reellen Übertragungsfunktion bezieht [105], [106], [107], [108]. Passive Systeme sind immer stabil und das Konzept kann verwendet werden, um nichtlineare Rückkopplungssysteme durch ein Regelungsgesetz asymptotisch zu stabilisieren, weshalb eine solche Systembeschreibung äußerst wünschenswert ist. Ein System

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \mathbf{u}), \quad (3.71a)$$

$$\mathbf{y} = \mathbf{c}(\mathbf{x}, \mathbf{u}) \quad (3.71b)$$

mit  $\mathbf{x} \in \mathbb{R}^n$ ,  $\mathbf{u}, \mathbf{y} \in \mathbb{R}^p$ ,  $\mathbf{f} : \mathbb{R}^n \times \mathbb{R}^p \rightarrow \mathbb{R}^n$ ,  $\mathbf{c} : \mathbb{R}^n \times \mathbb{R}^p \rightarrow \mathbb{R}^p$  ist passiv, falls eine stetig differenzierbare, positiv semidefinite<sup>19</sup> Energiefunktion  $V : \mathbb{R}^n \rightarrow \mathbb{R}$  mit

$$V(\mathbf{x}(t)) - V(\mathbf{x}(0)) \leq \int_0^t \mathbf{u}^\top \mathbf{y} \, d\tau \quad (3.72)$$

für alle  $\mathbf{x}, \mathbf{u}$  existiert. Aus Gleichung (3.72) folgt die differentielle Passivitätsungleichung

$$\dot{V}(\mathbf{x}(t)) = \frac{\partial V}{\partial \mathbf{x}} \dot{\mathbf{x}} \leq \mathbf{u}^\top \mathbf{y}, \quad (3.73)$$

sodass für die Energiebilanz des Systems gilt

$$\text{gespeicherte Energie} \leq \text{zugeführte Energie}. \quad (3.74)$$

Die Eigenschaft der Passivität ist von den Schnittstellen zur Umgebung abhängig, d. h. von der Wahl des Systemeingangs und -ausgangs [109]. Ein passives System mit positiv definiter Speicherfunktion  $V$  besitzt in  $\mathbf{x} = \mathbf{0}$  eine Ruhelage, die stabil im Sinne von Ljapunov ist. Für die Passivität verbundener Systeme gilt, dass die Parallelschaltung und die Rückkopplung (streng) passiver Systeme wiederum (streng) passiv ist. Diese Eigenschaft ist für den Regelungsentwurf ein außerordentlich nützliches Werkzeug [21].

Eine spezielle intrinsische Hamiltonsche Formulierung führt zu sogenannten *port-Hamiltonian* bzw. *port-controlled Hamiltonian systems* (PCH-Systeme) [110], [111],

<sup>19</sup>Eine Funktion  $f(\mathbf{x})$  heißt positiv semidefinit, wenn  $f(\mathbf{x}) \geq 0$  und  $f(\mathbf{0}) = 0$  [21].

[112], [113]. Erweitert um Dissipationseffekte ergeben sich sogenannte *port-controlled Hamiltonian systems with dissipation* (PCHD-Systeme) [114]. Die allgemeine Beschreibung solcher PCHD-Systeme ist durch

$$\dot{\mathbf{x}} = (\mathbf{J}(\mathbf{x}) - \mathbf{D}(\mathbf{x})) \left( \frac{\partial V}{\partial \mathbf{x}} \right)^\top + \mathbf{B}(\mathbf{x}) \mathbf{u}, \quad (3.75a)$$

$$\mathbf{y} = \mathbf{B}^\top(\mathbf{x}) \left( \frac{\partial V}{\partial \mathbf{x}} \right)^\top \quad (3.75b)$$

gegeben, wobei  $\mathbf{x} \in \mathbb{R}^n$  der Zustandsvektor und  $\mathbf{u}, \mathbf{y} \in \mathbb{R}^p$  die Ein- und Ausgangsschnittstellen des Systems sind.  $V : \mathbb{R}^n \rightarrow \mathbb{R}$  ist eine stetige positiv definite Funktion, die die gespeicherte Energie im System angibt.  $\mathbf{J}(\mathbf{x}) \in \mathbb{R}^{n \times n}$  ist eine schiefsymmetrische Matrix, d. h.  $\mathbf{J}(\mathbf{x}) = -\mathbf{J}^\top(\mathbf{x})$ , die die Energieflüsse im Inneren des Systems definiert und  $\mathbf{D}(\mathbf{x}) \in \mathbb{R}^{n \times n}$  ist eine positiv definite symmetrische Matrix, d. h.  $\mathbf{D}(\mathbf{x}) = \mathbf{D}^\top(\mathbf{x})$ , die das Verhalten der dissipativen Effekte abbildet<sup>20</sup>.  $\mathbf{B}(\mathbf{x})$  beschreibt den Energieaustausch des Systems mit der Systemumgebung. Die zeitliche Ableitung der Speicherfunktion ergibt

$$\dot{V}(\mathbf{x}) = \frac{\partial V}{\partial \mathbf{x}} \dot{\mathbf{x}} = \underbrace{\frac{\partial V}{\partial \mathbf{x}} \mathbf{J}(\mathbf{x}) \left( \frac{\partial V}{\partial \mathbf{x}} \right)^\top}_{=0, \text{ weil } \mathbf{J}(\mathbf{x}) = -\mathbf{J}^\top(\mathbf{x})} - \frac{\partial V}{\partial \mathbf{x}} \mathbf{D}(\mathbf{x}) \left( \frac{\partial V}{\partial \mathbf{x}} \right)^\top + \frac{\partial V}{\partial \mathbf{x}} \mathbf{B}(\mathbf{x}) \mathbf{u} \quad (3.76a)$$

$$\Rightarrow \dot{V}(\mathbf{x}) = \mathbf{u}^\top \mathbf{y} - \frac{\partial V}{\partial \mathbf{x}} \mathbf{D}(\mathbf{x}) \left( \frac{\partial V}{\partial \mathbf{x}} \right)^\top \leq \mathbf{u}^\top \mathbf{y}, \quad (3.76b)$$

sodass mit Gleichung (3.75a) die Passivitätsungleichung (3.73) erfüllt ist [21]. Demnach sind PCHD-Systeme immer passiv und somit immer stabil.

### 3.4.2 Algorithmus zur Bestimmung datengetriebener PCHD-Modelle

Für die datengetriebene Bestimmung solcher PCHD-Modelle unter gezielter Ausnutzung von physikalischem Vorwissen, vgl. [102], sind die nachstehenden Voraussetzungen zu erfüllen:

- Es werden zeitkontinuierliche eingangsaffine Systeme

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}) + \mathbf{B}\mathbf{u} \quad (3.77a)$$

$$\mathbf{y} = \mathbf{c}(\mathbf{x}, \mathbf{u}) \quad (3.77b)$$

mit  $\dim \mathbf{u} = \dim \mathbf{y}$  betrachtet.

- Es sind Daten des Zustands  $\mathbf{x}$  und des Systemeingangs  $\mathbf{u}$  verfügbar, die sowohl aus Messungen als auch aus Simulationen stammen können.
- Grundlegendes physikalisches Vorwissen über die gespeicherte Energie der zu modellierenden Dynamik ist bekannt.

<sup>20</sup>Wenn  $\mathbf{D}(\mathbf{x}) = \mathbf{0}$  ist, dann handelt es sich um ein (verlustfreies) PCH-System.

### 3 Modellbildung

Die Matrizen  $\mathbf{J}$ ,  $\mathbf{D}$  und  $\mathbf{B}$ , die im Allgemeinen von  $\mathbf{x}$  abhängig sind, vgl. Gleichung (3.75a), werden als konstant angenommen<sup>21</sup>. Außerdem werden die Matrizen  $\mathbf{J}$  und  $\mathbf{D}$  zusammengefasst zu  $\mathbf{K} = \mathbf{J} - \mathbf{D}$ . Diese Annahme stellt keine Einschränkung dar, weil sich jede quadratische Matrix eindeutig in einen symmetrischen und einen schiefsymmetrischen Teil zerlegen lässt

$$\mathbf{J} = \frac{\mathbf{K} - \mathbf{K}^\top}{2}, \quad \mathbf{D} = -\frac{\mathbf{K} + \mathbf{K}^\top}{2}. \quad (3.78)$$

Die Messdaten sind angeordnet in den Snapshot-Matrizen

$$\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_M] \in \mathbb{R}^{n \times M}, \quad (3.79a)$$

$$\mathbf{U} = [\mathbf{u}_1, \mathbf{u}_2, \dots, \mathbf{u}_M] \in \mathbb{R}^{p \times M}, \quad (3.79b)$$

$$\dot{\mathbf{X}} = [\dot{\mathbf{x}}_1, \dot{\mathbf{x}}_2, \dots, \dot{\mathbf{x}}_M] \in \mathbb{R}^{n \times M}. \quad (3.79c)$$

Physikalisches Vorwissen über die im System gespeicherte Energie

$$V(\mathbf{x}) = E_{\text{kinetisch}} + E_{\text{potentiell}} \quad (3.80)$$

wird für die Konstruktion der Funktion<sup>22</sup>

$$\Psi : \mathbb{R}^n \rightarrow \mathbb{R}^n, \quad \Psi(\mathbf{x}) = \left( \frac{\partial V}{\partial \mathbf{x}} \right)^\top \quad (3.81)$$

verwendet. Damit ergibt sich nach der Idee von Gleichung (3.75a) die Systembeschreibung

$$\dot{\mathbf{x}} = \mathbf{K}\Psi(\mathbf{x}) + \mathbf{B}\mathbf{u}. \quad (3.82)$$

Ähnlich wie beim EDMD-Verfahren wird der Zusammenhang über die Messdaten

$$\dot{\mathbf{X}} \approx \mathbf{K}\Psi(\mathbf{X}) + \mathbf{B}\mathbf{U} = [\mathbf{K}, \mathbf{B}] \begin{bmatrix} \Psi(\mathbf{X}) \\ \mathbf{U} \end{bmatrix} \quad (3.83)$$

betrachtet, sodass sich die Kleinste-Quadrate-Lösung mit

$$[\mathbf{K}, \mathbf{B}] = \dot{\mathbf{X}} \begin{bmatrix} \Psi(\mathbf{X}) \\ \mathbf{U} \end{bmatrix}^+ \quad (3.84)$$

ergibt. Als Nächstes werden die Matrizen  $\mathbf{J}$  und  $\mathbf{D}$  mittels Gleichung (3.78) berechnet. Um eine PCHD-Form zu erreichen, vgl. Gleichung (3.75a), ist es erforderlich, dass  $\mathbf{D}$  positiv semidefinit ist. Die symmetrische Matrix  $\mathbf{D}$  ist genau dann positiv semidefinit, wenn alle Eigenwerte von  $\mathbf{D}$  größer oder gleich null sind. Die im Sinne der Frobeniusnorm nächste Projektion der Matrix  $\mathbf{D}$  in die Menge der positiv

<sup>21</sup>Diese Annahme beruht darauf, dass sich die Dynamik eines PCHD-Modells näherungsweise durch konstante Matrizen beschreiben lässt, ähnlich der Vorgehensweise bei der Vereinfachung eines nichtlinearen Modells durch Linearisierung um einen oder mehrere Arbeitspunkte.

<sup>22</sup>Der Gradient der Energiefunktion  $V$  wird hier aufgrund des verwandten Algorithmus in Analogie zu den Observablen beim EDMD-Verfahren mit  $\Psi(\mathbf{x})$  bezeichnet.

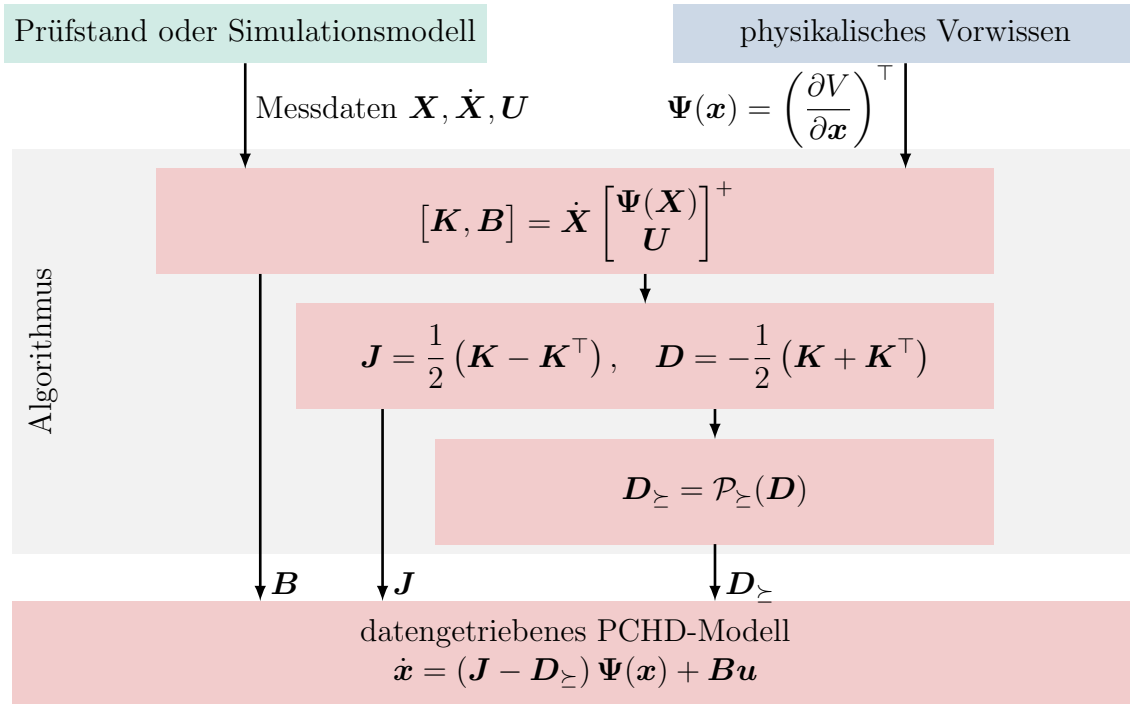
semidefiniten Matrizen erfolgt mittels

$$\mathbf{D}_{\succeq} = \mathcal{P}_{\succeq}(\mathbf{D}) = \mathbf{V} \max(\mathbf{0}_{n \times n}, \mathbf{\Sigma}) \mathbf{V}^{\top} := \mathbf{V} \begin{bmatrix} \max(0, \lambda_1) & & 0 \\ & \ddots & \\ 0 & & \max(0, \lambda_n) \end{bmatrix}, \quad (3.85)$$

wobei  $\mathbf{V} \mathbf{\Sigma} \mathbf{V}^{\top}$  die orthogonale Eigenzerlegung von  $\mathbf{D}$  ist. Die resultierende Systembeschreibung liegt in PCHD-Form vor und ist passiv

$$\dot{\mathbf{x}} = (\mathbf{J} - \mathbf{D}_{\succeq}) \Psi(\mathbf{x}) + \mathbf{B} \mathbf{u}. \quad (3.86)$$

Der Algorithmus ist in der Abbildung 3.22 zusammengefasst.



**Abbildung 3.22** Schematische Übersicht der Methode zur Bestimmung eines datengetriebenen PCHD-Modells.

Mittels der beschriebenen Methode lassen sich **datengetriebene PCHD-Modelle** von eingangsaffinen Systemen mit konstanter Eingangsmatrix unter Nutzung von physikalischem Vorwissen über die im System gespeicherte Energie bestimmen. Es werden Messdaten des **vollständigen** Zustandsvektors und dessen zeitliche Ableitungen benötigt. Die resultierenden **zeitkontinuierlichen** Modelle weisen eine hohe Modellgenauigkeit auf und sind gleichzeitig aufgrund der speziellen Struktur hochgradig physikalisch interpretierbar.

### 3.4.3 Beispielhafte Anwendung

Im Folgenden wird die Anwendbarkeit der datengetriebenen PCHD-Modelle Verfahren beispielhaft hinsichtlich der erreichbaren Prädiktionsgüte demonstriert. Hierfür

wird das numerisch simulierte nichtlineare Verhalten des jeweiligen Originalsystems mit dem approximierten PCHD-Modell verglichen. Als simulative Beispielsysteme werden das im Abschnitt 2.1 eingeführte Einfachpendel sowie ein Doppelpendel und ein vereinfachter Multicopter betrachtet. Als experimentelles Beispielsystem wird der Schlagmechanismus des institutseigenen Golfroboters betrachtet, dessen Dynamik nichtlineare Reibungseffekte enthält.

**Pendel** Das resultierende datengetriebene PCHD-Modell für das Pendel (2.1) sowie eine ausführliche Beschreibung des Systems und der verwendeten Trainingsdaten wurden bereits im Abschnitt 2.1 vorab präsentiert. Aufgrund der im System gespeicherte Energie

$$V(\mathbf{x}) = \frac{1}{2}ml^2\dot{x}_2^2 + mgl(1 - \cos x_1) \quad (3.87)$$

ergibt sich

$$\Psi(\mathbf{x}) = \left( \frac{\partial V}{\partial \mathbf{x}} \right)^\top = \begin{bmatrix} mgl \sin x_1 \\ ml^2 \dot{x}_2 \end{bmatrix}. \quad (3.88)$$

Der Algorithmus in der Abbildung 3.22 liefert das datengetriebene PCHD-Modell mit

$$\mathbf{J} = \begin{bmatrix} 0 & 4 \\ -4 & 0 \end{bmatrix} = -\mathbf{J}^\top, \quad \mathbf{D} = \begin{bmatrix} 0 & 0 \\ 0 & 1,6 \end{bmatrix} = \mathbf{D}^\top \succeq 0, \quad \mathbf{b} = \begin{bmatrix} 0 \\ 4 \end{bmatrix} \quad (3.89)$$

und damit

$$\begin{aligned} \begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} &= (\mathbf{J} - \mathbf{D}) \left( \frac{\partial V}{\partial \mathbf{x}} \right)^\top + \mathbf{b}u \\ &= \left( \begin{bmatrix} 0 & 4 \\ -4 & 0 \end{bmatrix} - \begin{bmatrix} 0 & 0 \\ 0 & 1,6 \end{bmatrix} \right) \begin{bmatrix} mgl \sin x_1 \\ ml^2 \dot{x}_2 \end{bmatrix} + \begin{bmatrix} 0 \\ 4 \end{bmatrix} u. \end{aligned} \quad (3.90)$$

Diese Matrizen korrespondieren exakt mit der analytisch hergeleiteten PCHD-Form des Originalsystems (2.1)

$$\begin{aligned} \begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} &= \begin{bmatrix} -\frac{g}{l} \sin x_1 - \frac{d}{ml^2} x_2 + \frac{1}{ml^2} u \\ \frac{1}{ml^2} x_2 \end{bmatrix} \\ &= \left( \begin{bmatrix} 0 & \frac{1}{ml^2} \\ -\frac{1}{ml^2} & 0 \end{bmatrix} - \begin{bmatrix} 0 & 0 \\ 0 & \frac{d}{ml^2 l^4} \end{bmatrix} \right) \begin{bmatrix} mgl \sin x_1 \\ ml^2 \dot{x}_2 \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{1}{ml^2} \end{bmatrix} u \end{aligned} \quad (3.91)$$

mit

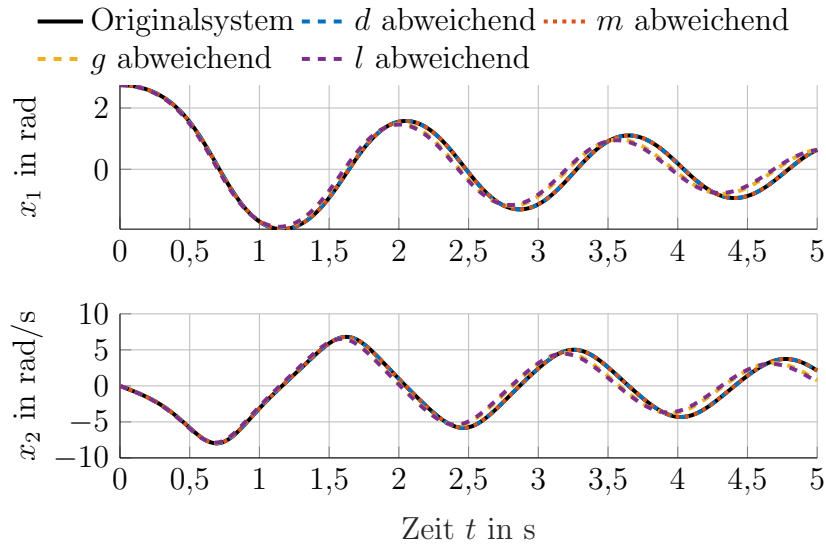
$$\mathbf{J}_{\text{phys}} = \begin{bmatrix} 0 & \frac{1}{ml^2} \\ -\frac{1}{ml^2} & 0 \end{bmatrix}, \quad \mathbf{D}_{\text{phys}} = \begin{bmatrix} 0 & 0 \\ 0 & \frac{d}{ml^2 l^4} \end{bmatrix}, \quad \mathbf{b}_{\text{phys}} = \begin{bmatrix} 0 \\ \frac{1}{ml^2} \end{bmatrix}. \quad (3.92)$$

Die Berechnung des Algorithmus liefert unmittelbar  $\mathbf{D}_\succeq = \mathbf{D}$ , sodass keine nachträgliche Verschiebung der Eigenwerte erforderlich ist.

Zur Untersuchung der Robustheit gegenüber Modellunsicherheiten bzw. fehlerhaftem physikalischen Vorwissen wurden datengetriebene PCHD-Modelle für das Pendel bestimmt, bei denen für die Festlegung von  $\Psi(\mathbf{x})$  eine Abweichung einzelner Parameter um  $\pm 10\%$  vom Originalwert angenommen wurde. Die Abbildung 3.23 zeigt die dazugehörigen prädizierten Verläufe für das autonom schwingende Pendel.



Es ist zu erkennen, dass die Verläufe trotz der Parameterabweichungen von  $\pm 10\%$  nur geringfügig vom Originalsystem abweichen. Die Abweichungen der Parameter  $m$  und  $d$  werden durch den datengetriebenen Algorithmus vollständig korrigiert. Die Ergebnisse für die Abweichungen der Parameter  $g$  und  $l$  lassen sich hingegen dadurch erklären, dass diese Parameter einen erheblichen Einfluss auf die Schwingungsdynamik, d. h. die Eigenfrequenz, haben. Die hier angenommene Abweichung dient illustrativen Zwecken; bei vorhandener Unsicherheit bezüglich der Systemparameter ist es im Allgemeinen empfehlenswert, diese im Vorfeld zu identifizieren oder alternativ in Form einer Hyperparameteroptimierung zu überlagern.



**Abbildung 3.23** Simulative Analyse fehlerhaft gewählter physikalischer Parameter anhand identifizierter PCHD-Modelle mit Parameterabweichungen von  $\pm 10\%$ .

**Schlagmechanismus des Golfroboters** Der Golfroboter, vgl. Abbildung 6.1, wird im Abschnitt 6.1 ausführlich beschrieben. Die Dynamik des Schlagmechanismus ergibt sich durch

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} x_2 \\ \frac{-mga \sin x_1 - M_d(x) + 4u}{J} \end{bmatrix}, \quad (3.93a)$$

$$M_d(\mathbf{x}) = dx_2 + r\mu \operatorname{sgn} x_2 |mx_2^2 a + mg \cos x_1|, \quad (3.93b)$$

$$y = x_1. \quad (3.93c)$$

Die im System gespeicherte Energie ist gegeben durch

$$V(\mathbf{x}) = \frac{1}{2} J x_2^2 + mga(1 - \cos x_1), \quad (3.94)$$

sodass

$$\Psi(\mathbf{x}) = \left( \frac{\partial V}{\partial \mathbf{x}} \right)^\top = \begin{bmatrix} mga \sin x_1 \\ J x_2 \end{bmatrix}. \quad (3.95)$$

### 3 Modellbildung

Der Algorithmus in der Abbildung 3.22 liefert für dieselben Trainingsdaten wie in der Abbildung 3.11 das datengetriebene PCHD-Modell

$$\mathbf{J} = \begin{bmatrix} 0 & 6,1844 \\ -6,1844 & 0 \end{bmatrix}, \quad \mathbf{b} = \begin{bmatrix} 0 \\ 22,9965 \end{bmatrix}, \quad (3.96a)$$

$$\mathbf{D} = \begin{bmatrix} 0 & -0,736 \\ -0,736 & 6,4364 \end{bmatrix} \text{ mit den Eigenwerten } \lambda_1 = -0,0831, \lambda_2 = 6,5195 \quad (3.96b)$$

und der positiv semidefiniten Matrix

$$\mathbf{D}_\succeq = \begin{bmatrix} 0,0820 & -0,7267 \\ -0,7267 & 6,4375 \end{bmatrix} \text{ mit den Eigenwerten } \lambda_1 = 0, \lambda_2 = 6,5195. \quad (3.97)$$

Das physikalische PCHD-Modell

$$\mathbf{J}_{\text{phys}} = \begin{bmatrix} 0 & \frac{1}{J} \\ -\frac{1}{J} & 0 \end{bmatrix} \approx \begin{bmatrix} 0 & 6,9204 \\ -6,9204 & 0 \end{bmatrix}, \quad \mathbf{D}_{\text{phys}}(\mathbf{x}) = \begin{bmatrix} 0 & 0 \\ 0 & d_{\text{phys}}(\mathbf{x}) \end{bmatrix}, \quad (3.98a)$$

$$\mathbf{b}_{\text{phys}} = \begin{bmatrix} 0 \\ \frac{4}{J} \end{bmatrix} \approx \begin{bmatrix} 0 \\ 27,6817 \end{bmatrix} \quad (3.98b)$$

$$\text{mit } d_{\text{phys}}(\mathbf{x}) = \begin{cases} \frac{d}{J^2}, & x_2 = 0 \\ \frac{d}{J^2} + \frac{r\mu}{J^2} \left| \frac{mx_2^2 a + mg \cos x_1}{x_2} \right|, & x_2 \neq 0 \end{cases} \quad (3.98c)$$

lässt sich analytisch aus dem physikalischen nichtlinearen Modell (3.93) herleiten.

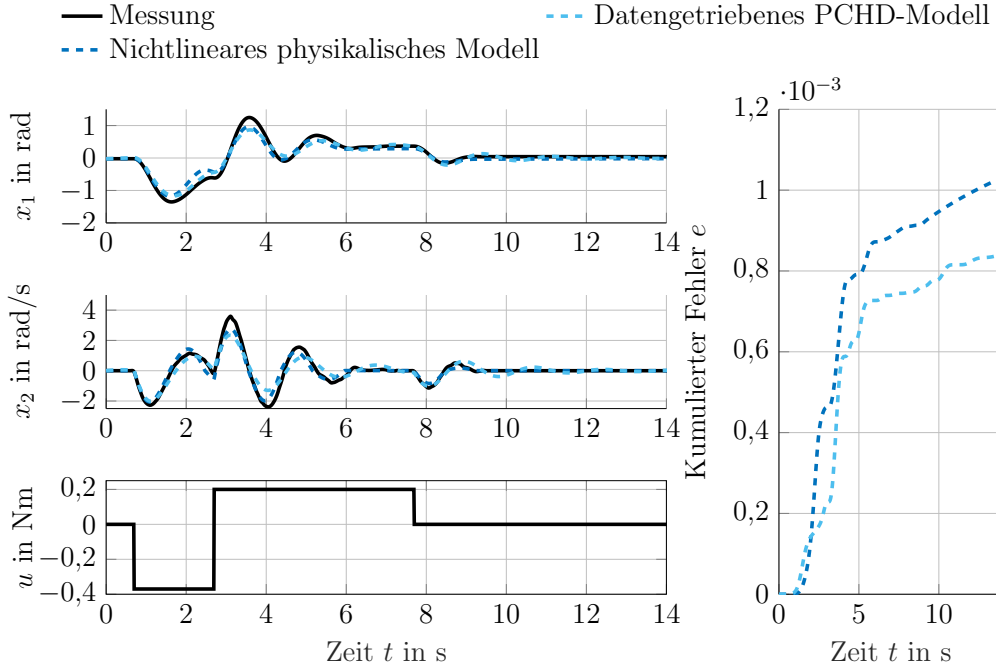
An dieser Stelle ist anzumerken, dass beim Schlagmechanismus des Golfroboters  $\mathbf{D}_{\text{phys}}(\mathbf{x})$  von  $\mathbf{x}$  abhängt. Angesichts der dennoch hohen Modellgüte des datengetriebenen PCHD-Modells mit konstanter Matrix  $\mathbf{D}$ , vgl. Abbildung 3.24, kann davon ausgegangen werden, dass die dominanten Nichtlinearitäten bereits in der Energiefunktion  $V(\mathbf{x})$  berücksichtigt sind.

**Doppelpendel** Anhand eines unaktuerten und reibungsfreien Doppelpendels, vgl. Skizze in der Abbildung 3.25, werden die Grenzen des vorgestellten Verfahrens zur datengetriebenen Bestimmung von PCHD-Modellen untersucht. Dieses System zeigt eine chaotische Dynamik. Das bedeutet, dass selbst minimale Änderungen der Anfangsbedingungen eines chaotischen Systems zu stark unterschiedlichen Trajektorien führen können, vgl. [115]. Im Folgenden wird zunächst kurz die physikalisch motivierte Modellbildung beschrieben. Anschließend wird die datengetriebene Bestimmung eines PCHD-Modells diskutiert.

Ein physikalisch motiviertes nichtlineares Modell lässt sich mittels des Lagrange-Formalismus herleiten<sup>23</sup>. Unter der Annahme, dass die Nulllage  $\varphi = 0$  sich in der unteren Ruhelage befindet und die Pendelarme als Punktmassen mit den in der Tabelle 3.1 angegebenen Parametern modelliert sind, ergeben sich die kinetische und potentielle Energie des Systems durch

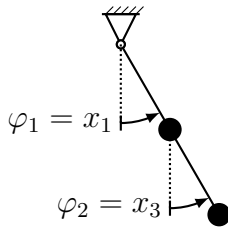
$$E_{\text{kinetisch}} = ml^2 \left( \dot{\varphi}_1^2 + \frac{1}{2} \dot{\varphi}_2^2 + \dot{\varphi}_1 \dot{\varphi}_2 \cos(\varphi_1 - \varphi_2) \right), \quad (3.99a)$$

<sup>23</sup>Eine ausführliche Herleitung der Bewegungsdifferentialgleichungen findet sich beispielsweise in [116].



**Abbildung 3.24** Das datengetriebene PCHD-Modell weist eine höhere Prädiktionsgüte als das nichtlineare physikalische Modell (3.93) auf.

$$E_{\text{potentiell}} = -mgl(2 \cos \varphi_1 + \cos \varphi_2). \quad (3.99b)$$



**Abbildung 3.25**  
Doppelpendel

**Tabelle 3.1** Physikalische Parameter

Symbol	Physikalischer Parameter	Wert
$m_1 = m_2 = m$	(Punkt-)Massen der Pendel	1 kg
$l_1 = l_2 = l$	Längen der beiden Pendelarme	0,5 m
$g$	Gravitationsbeschleunigung	$9,81 \text{ m s}^{-2}$

Für die Lagrange-Funktion gilt damit

$$\begin{aligned} \mathcal{L} &= E_{\text{kinetisch}} - E_{\text{potentiell}} \\ &= ml^2 \left( \dot{\varphi}_1^2 + \frac{1}{2} \dot{\varphi}_2^2 + \dot{\varphi}_1 \dot{\varphi}_2 \cos(\varphi_1 - \varphi_2) \right) + mgl(2 \cos \varphi_1 + \cos \varphi_2), \end{aligned} \quad (3.100)$$

sodass sich die Bewegungsdifferentialgleichungen formuliert in den Minimalkoordinaten  $[q_1, q_2]^\top = [\varphi_1, \varphi_2]^\top$  mittels des Lagrange-Formalismus

$$\frac{d}{dt} \frac{\partial \mathcal{L}}{\partial \dot{q}_i} - \frac{\partial \mathcal{L}}{\partial q_i} = 0 \quad (3.101)$$

### 3 Modellbildung

zu

$$ml^2 (2\ddot{\varphi}_1 + \ddot{\varphi}_2 \cos(\varphi_1 - \varphi_2) + \dot{\varphi}_2^2 \sin(\varphi_1 - \varphi_2)) + 2mlg \sin \varphi_1 = 0, \quad (3.102a)$$

$$ml^2 (\ddot{\varphi}_2 + \ddot{\varphi}_1 \cos(\varphi_1 - \varphi_2) - \dot{\varphi}_1^2 \sin(\varphi_1 - \varphi_2)) + mlg \sin \varphi_2 = 0 \quad (3.102b)$$

ergeben. Mit  $[x_1, x_2, x_3, x_4]^\top = [\varphi_1, \dot{\varphi}_1, \varphi_2, \dot{\varphi}_2]^\top$  lässt sich die Dynamik des Doppelpendels im Zustandsraum formulieren:

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \\ \dot{x}_4 \end{bmatrix} = \begin{bmatrix} x_2 \\ \frac{\frac{g}{l}(\sin x_3 \cos(x_1 - x_3) - 2 \sin x_1) - \sin(x_1 - x_3)(x_4^2 + x_2^2 \cos(x_1 - x_3))}{2 - \cos^2(x_1 - x_3)} \\ x_4 \\ \frac{2\frac{g}{l}(\sin x_1 \cos(x_1 - x_3) - \sin x_3) + \sin(x_1 - x_3)(2x_2^2 + x_4^2 \cos(x_1 - x_3))}{2 - \cos^2(x_1 - x_3)} \end{bmatrix}. \quad (3.103)$$

Für das datengetriebene PCHD-Modell wird physikalisches Vorwissen in Form der im System gespeicherten Energie

$$\begin{aligned} V(\mathbf{x}) &= E_{\text{kinetisch}} + E_{\text{potentiell}} \\ &= ml^2 \left( x_2^2 + \frac{1}{2} x_4^2 + x_2 x_4 \cos(x_1 - x_3) \right) - mlg (2 \cos x_1 + \cos x_3) \end{aligned} \quad (3.104)$$

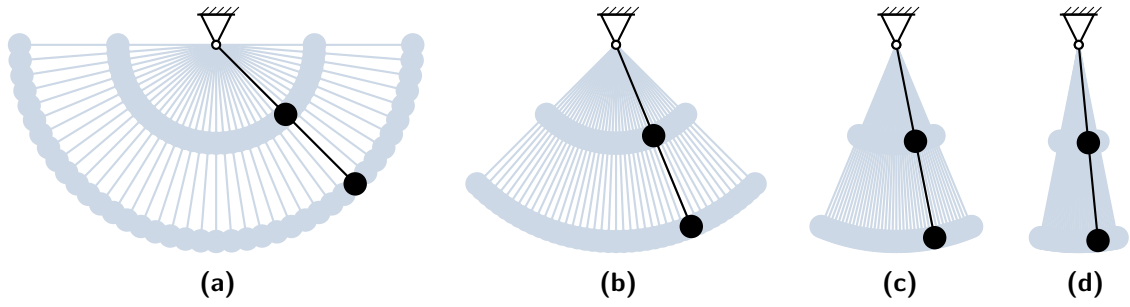
benötigt. Der Gradient von  $V(\mathbf{x})$  definiert die Funktion

$$\Psi(\mathbf{x}) = \left( \frac{\partial V}{\partial \mathbf{x}} \right)^\top = \begin{bmatrix} -ml^2 x_2 x_4 \sin(x_1 - x_3) + 2mlg \sin x_1 \\ 2ml^2 x_2 + ml^2 x_4 \cos(x_1 - x_3) \\ ml^2 x_2 x_4 \sin(x_1 - x_3) + mlg \sin x_3 \\ ml^2 x_4 + ml^2 x_2 \cos(x_1 - x_3) \end{bmatrix}. \quad (3.105)$$

Für die Analyse der Modellgenauigkeit des datengetriebenen PCHD-Modells werden unterschiedliche Szenarien betrachtet, bei denen die Trainingsdaten sowie die Testtrajektorie variiert werden, vgl. Abbildung 3.26. Für jedes Szenario werden 100 Trajektorien mit einer Dauer von jeweils 3s und einem additiven weißen Rauschen simulativ generiert. Die betrachteten Anfangsauslenkungen werden vom Szenario in Abbildung 3.26(a) bis zum Szenario in Abbildung 3.26(d) schrittweise reduziert, wobei die Zahlenwerte der Matrizen der berechneten datengetriebenen PCHD-Modelle in der Dissertation [16] dargestellt sind. In der Abbildung 3.27 ist jeweils der kumulierte Fehler

$$e(t_k) = \sum_{m=1}^k \frac{1}{k} \|\mathbf{x}_{\text{Originalsystem}}(t_m) - \mathbf{x}_{\text{Prädiktion}}(t_m)\|^2 \quad (3.106)$$

der unterschiedlichen datengetriebenen PCHD-Modelle dargestellt. Es ist zu erkennen, dass alle vier datengetriebenen PCHD-Modelle in der Lage sind, die chaotische Dynamik des Doppelpendels zumindest für 0,4s präzise zu prädictieren. Mit steigender Anfangsauslenkung und für einen längeren Zeithorizont sinkt die Modellgenauigkeit jedoch stark. Eine ähnliche Beobachtung wurde im Beitrag [117] anhand des EDMD-Verfahrens formuliert, dessen Modellgenauigkeit sich mit steigender im System gespeicherter Energie reduziert.



**Abbildung 3.26** Das Doppelpendel dient dazu die Grenzen des vorgestellten Verfahrens zur datengetriebenen Bestimmung von PCHD-Modellen zu illustrieren. Es werden vier unterschiedliche Szenarien der Trainings- und Testdaten untersucht. Die Anfangsauslenkungen für die Trainingsdaten sind schematisch in blau und die Anfangsauslenkung für die Testtrajektorie in schwarz dargestellt.

Aus der Betrachtung von Gleichung (3.103) und Gleichung (3.105) ist ersichtlich, dass es für das Doppelpendel nicht ohne Weiteres gelingt, eine PCHD-Beschreibung analytisch herzuleiten. Darüber hinaus wäre die Matrix  $\mathbf{J}_{\text{phys}}(\mathbf{x})$  von  $\mathbf{x}$  abhängig. Die Approximation der Dynamik des Doppelpendels mittels einer konstanten Matrix  $\mathbf{J}$  lässt sich hier ähnlich interpretieren wie die Linearisierung der Dynamik um den Betriebspunkt der unteren Ruhelage. Diese Schlussfolgerung deckt sich mit der Analyse der Modellgenauigkeit für unterschiedliche Anfangsauslenkungen, vgl. Abbildung 3.27. Am Beispiel des Doppelpendels erreicht das im Abschnitt 3.4 beschriebene Verfahren zur datengetriebenen Bestimmung von PCHD-Modellen für größere Anfangsauslenkungen die Grenze der erreichbaren Modellgenauigkeit.

**Zweidimensionaler Multicopter** Als weiteres Beispiel für die Anwendung datengetriebener PCHD-Modelle dient ein zweidimensionaler Multicopter. Multicopter sind Luftfahrzeuge, deren Auftrieb durch mehrere<sup>24</sup> nach unten wirkende Propeller erzeugt wird, wodurch ein senkrechtes Starten und Landen ermöglicht wird. Durch die Variation der Propellerdrehzahlen lässt sich die Orientierung und damit das Flugverhalten gezielt steuern [118]. Aufgrund der Flexibilität, Effizienz und Zuverlässigkeit bei gleichzeitig niedrigen Wartungskosten spielen Multicopter eine herausragende Rolle beispielsweise bei der Paketzustellung [119] oder in der Katastrophenhilfe [120].

Der Einfachheit halber wird an dieser Stelle ein Multicopter mit zwei Propellern in der Ebene betrachtet, vgl. Skizze in der Abbildung 3.28. In Anlehnung an [121], [122] ergeben sich die Differentialgleichungen der als linear gedämpft angenommenen Flugbewegung in drei Freiheitsgraden mit den Parametern aus der Tabelle 3.2 zu:

$$m\ddot{x} = -(F_1 + F_2) \sin \varphi - d_t \dot{x}, \quad (3.107a)$$

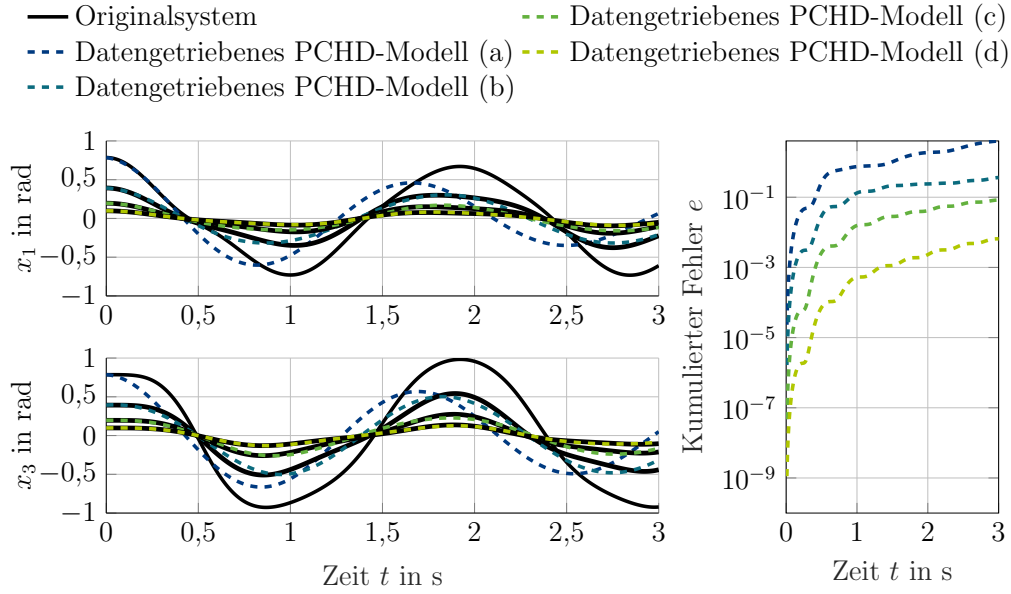
$$m\ddot{y} = (F_1 + F_2) \cos \varphi - mg - d_t \dot{y}, \quad (3.107b)$$

$$J\ddot{\varphi} = (F_2 - F_1) l - d_r \dot{\varphi}. \quad (3.107c)$$

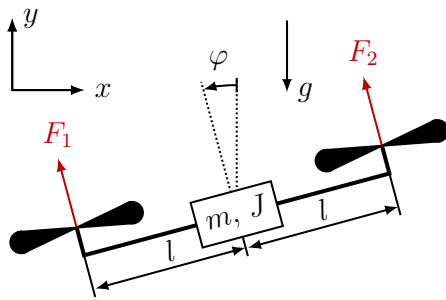
Mit  $[x_1, x_2, x_3, x_4, x_5, x_6]^\top = [x, \dot{x}, y, \dot{y}, \varphi, \dot{\varphi}]^\top$  und  $[u_1, u_2]^\top = [F_1, F_2]^\top$  lässt sich

<sup>24</sup>Multicopter mit vier Propellern werden beispielsweise als Quadrocopter, solche mit sechs Propellern als Hexacopter bezeichnet.

### 3 Modellbildung



**Abbildung 3.27** Die Prädiktionsgüte des datengetriebenen PCHD-Modells für das simulierte Doppelpendel sinkt bei steigender Anfangsauslenkung stark. Aus Platzgründen wird auf die Darstellung der Verläufe für die Zustände  $x_2$  und  $x_4$  verzichtet.



**Abbildung 3.28** Multicopter.

**Tabelle 3.2** Physikalische Parameter.

Symbol	Physikalischer Parameter	Wert
$m$	Masse	1 kg
$J$	Rotationsträgheit	$0,5 \text{ kg m}^2$
$l$	Länge zu den Propellern	0,5 m
$g$	Gravitationsbeschleunigung	$9,81 \text{ m s}^{-2}$
$d_t$	translatorische Dämpfung	$0,1 \text{ kg s}^{-1}$
$d_r$	rotatorische Dämpfung	$0,1 \text{ kg m}^2 \text{ s}^{-1}$

die Dynamik im Zustandsraum

$$\begin{aligned}
 \begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \\ \dot{x}_4 \\ \dot{x}_5 \\ \dot{x}_6 \end{bmatrix} &= \begin{bmatrix} x_2 \\ -\frac{1}{m}(u_1 + u_2) \sin x_5 - \frac{d_t}{m}x_2 \\ x_4 \\ \frac{1}{m}(u_1 + u_2) \cos x_5 - g - \frac{d_t}{m}x_4 \\ x_6 \\ \frac{l}{J}(u_2 - u_1) - \frac{d_r}{J}x_6 \end{bmatrix} \\
 &= \begin{bmatrix} x_2 \\ -\frac{d_t}{m}x_2 \\ x_4 \\ -g - \frac{d_t}{m}x_4 \\ x_6 \\ -\frac{d_r}{J}x_6 \end{bmatrix} + \underbrace{\begin{bmatrix} 0 & 0 \\ -\frac{1}{m} \sin x_5 & -\frac{1}{m} \sin x_5 \\ 0 & 0 \\ \frac{1}{m} \cos x_5 & \frac{1}{m} \cos x_5 \\ 0 & 0 \\ -\frac{l}{J} & \frac{l}{J} \end{bmatrix}}_{B(x)} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix}. \quad (3.108)
 \end{aligned}$$

formulieren. Das System ist eingangsaffin mit einer Eingangsmatrix  $\mathbf{B}(\mathbf{x})$ , die nicht konstant ist. Die im System gespeicherte Energie beträgt

$$V(\mathbf{x}) = E_{\text{kinetisch}} + E_{\text{potentiell}} = \frac{1}{2}m(x_2^2 + x_4^2) + \frac{1}{2}Jx_6^2 + mgx_3, \quad (3.109)$$

sodass der Gradient sich zu

$$\left(\frac{\partial V}{\partial \mathbf{x}}\right)^\top = [0, mx_2, mg, mx_4, 0, Jx_6]^\top \quad (3.110)$$

berechnet. Die analytisch hergeleitete PCHD-Darstellung ist gegeben durch

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \\ \dot{x}_4 \\ \dot{x}_5 \\ \dot{x}_6 \end{bmatrix} = \begin{bmatrix} 0 & \frac{1}{m} & 0 & 0 & 0 & 0 \\ -\frac{1}{m} & -\frac{d_t}{m^2} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{1}{m} & 0 & 0 \\ 0 & 0 & -\frac{1}{m} & -\frac{d_t}{m^2} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{1}{J} \\ 0 & 0 & 0 & 0 & -\frac{1}{J} & -\frac{d_t}{J^2} \end{bmatrix} \begin{bmatrix} 0 \\ mx_2 \\ mg \\ mx_4 \\ 0 \\ Jx_6 \end{bmatrix} + \mathbf{B}(\mathbf{x}) \begin{bmatrix} u_1 \\ u_2 \end{bmatrix}, \quad (3.111)$$

wobei die Elemente in der ersten und fünften Spalte der Matrix  $\mathbf{K}$  aufgrund der partiellen Ableitungen  $\psi_1(\mathbf{x}) = \frac{\partial V}{\partial x_1} = 0$ ,  $\psi_5(\mathbf{x}) = \frac{\partial V}{\partial x_5} = 0$  beliebig wählbar sind. Diese Eigenschaft führt dazu, dass der Algorithmus nicht in der Lage ist, die Matrizen  $\mathbf{J}$  und  $\mathbf{D}$  korrekt zu berechnen. Eine nahe liegende Lösung für dieses Problem könnte eine Modellordnungsreduktion des Systems auf die für die gespeicherte Energie relevanten Zustände

$$\tilde{\mathbf{x}} = [\tilde{x}_1, \tilde{x}_2, \tilde{x}_3, \tilde{x}_4]^\top = [\dot{x}, y, \dot{y}, \dot{\varphi}]^\top \quad (3.112)$$

darstellen. Dieser Ansatz ist bei dem betrachteten Beispiel jedoch nicht anwendbar, weil die Eingangsmatrix  $\mathbf{B}(\mathbf{x})$  vom Winkel  $\varphi$  abhängig ist und daher die Prädiktion des Winkels  $\varphi$  erforderlich ist.

### Automatische Modellaktualisierung

Eine Herausforderung mechatronischer Systeme liegt in vorhersehbaren oder unvorhersehbaren Systemveränderungen, z. B. die Installation eines neuen Bauteils, Verschleiß oder Temperaturschwankungen während des Betriebs. Um dennoch eine gleichbleibend hohe Regelungsgüte gewährleisten zu können, muss die Informationsverarbeitung in der Lage sein, diese Änderungen zu erfassen und sich automatisch anzupassen. Daher wurde in [123] eine Erweiterung um selbstlernende Streckenmodelle entwickelt, die sich auf Basis laufend aufgezeichneter Messdaten an veränderliches Systemverhalten anpassen, vgl. Abbildung 3.13. Dafür wurde zunächst in Anlehnung an [123] ein Algorithmus für ein adaptives PCHD-Modell mit einem rekursiven kleinste-Quadrate-Ansatz realisiert. Durch einen variablen Vergessensfaktor berücksichtigt das Modell vergangene Messdaten in Abhängigkeit der Intensität der auftretenden Systemveränderungen zu vernachlässigen. Die Begrenzung der Spur der Kovarianzmatrix verhindert das einhergehende Risiko eines sogenannten Kovarianz-Windups. Experimentelle Untersuchungen am Schlagmechanismus des Golfroboters

zeigten eindrucksvoll, dass das adaptive PCHD-Modell die nichtlineare Systemdynamik mit sehr hoher Genauigkeit abbilden kann.

Die erfolgreiche Integration der Adaptionfähigkeit bestätigt die hohe regelungstechnische Verwertbarkeit der hybriden PCHD-Modelle gemäß der Kriterien, die im Kapitel 2 formuliert wurden.

## 3.5 Probabilistische hybride Modellierung mittels Gauß-Prozess-Regression

In diesem Abschnitt wird eine hybride Modellierung vorgestellt, welche auf der sogenannten Gauß-Prozess-Regression basiert. Dabei wird der vorhandene Modellfehler des physikalischen Modells als Zufallsvariable definiert und über einen Gauß-Prozess nachgebildet. Auf der Grundlage von Messdaten wird der Fehler somit als normalverteilt angenommen und durch einen zustandsabhängigen Erwartungswert und eine Varianz dargestellt. Auf diese Weise lässt sich die Unsicherheit über den Modellfehler quantifizieren und insbesondere für lernende und sicherheitskritische Systeme verwenden. In Unterabschnitt 3.5.1 wird die Entwicklung des datengetriebenen Modellteils formal beschrieben. Anschließend erfolgt in Unterabschnitt 3.5.2 die Verbindung vom physikalischen und datengetriebenen Modellteil über die Unscented Transformation. Hierbei wird der Begriff der Zustandspropagation eingeführt, welche eine wahrscheinlichkeitsbasierte Langzeitprädiktion des hybriden Gesamtmodells ermöglicht.

### 3.5.1 Entwicklung des datengetriebenen Modellteils

Zunächst wird eine formale Unterscheidung zwischen den verschiedenen betrachteten Systemen eingeführt. Das reale System, welches beispielsweise als Prüfstand oder Prototyp vorhanden ist, wird mit  $f$  bezeichnet. Das unvollständige Modell, das in der Regel auf physikalischen Gesetzmäßigkeiten beruht, wird entsprechend einer Zustandsschätzung mit  $\hat{f}$  beschrieben - es handelt sich um eine Annäherung an  $f$ . Des Weiteren erhält der rein datengetriebene Modellteil, welcher dem Modellfehler zugeordnet wird, die Bezeichnung  $\Delta\mathbf{x}$ . Das gesamte hybride Modell wird mit  $\tilde{f}$  angegeben.

Auf dieser Basis lässt sich das reale diskrete System, unter der herkömmlichen Annahme einer additiven Struktur, mit

$$\mathbf{x}_{k+1} = \mathbf{f}(\mathbf{x}_k, \mathbf{u}_k) = \hat{\mathbf{f}}(\mathbf{x}_k, \mathbf{u}_k) + \Delta\mathbf{x}_{k+1}(\mathbf{x}_k, \mathbf{u}_k) \quad (3.113)$$

angeben. Nun ist das Ziel eine Beschreibung der Modellfehler  $\Delta\mathbf{x}_{k+1}(\mathbf{x}_k, \mathbf{u}_k)$  zu finden und ein hybrides Modell aufzustellen, um eine im Vergleich zum physikalischen Modell bessere Annäherung an das reale System zu erhalten. Ein wesentlicher Anhaltspunkt stellen die aufgenommenen Messdaten vom realen System bereit. Die zugehörigen Triple  $(\mathbf{x}_k, \mathbf{x}_{k+1}, \mathbf{u}_k)$ , welche aus dem aktuellen Zustand, dem Folgezustand und der Stellgröße bestehen, werden zeitlich gesehen in den Datenmatrizen

$$\mathbf{X}_k, \mathbf{X}_{k+1} \in \mathbb{R}^{n_x \times n_d}, \mathbf{U}_k \in \mathbb{R}^{n_u \times n_d} \quad (3.114)$$



gesammelt und zusammengefasst. Hierbei bezeichnet  $n_x$  die Dimension des Zustandsvektors und  $n_d$  die Anzahl von Messpunkten. Anhand des physikalischen Modells wird der Modellfehler entsprechend zu Gleichung (3.113) mit

$$\Delta \mathbf{X}_{k+1}^{(i)} = \mathbf{X}_{k+1}^{(i)} - \hat{\mathbf{f}}(\mathbf{X}_k^{(i)}, \mathbf{U}_k^{(i)}), \quad i = 1, \dots, n_d, \quad (3.115)$$

für jeden Messpunkt berechnet und in  $\Delta \mathbf{X}_{k+1}$  abgespeichert. Jede Dimension des Modellfehlers wird als parameterabhängige Zufallsvariable behandelt und über einen separaten Gauß-Prozess [124]

$$\Delta \mathbf{x}_{k+1}^{(j)} \sim \mathcal{GP}_j(0, k(\mathbf{x}_k^u, \mathbf{x}_k^{u'}; \boldsymbol{\eta}_j)), \quad j = 1, \dots, n_x, \quad (3.116)$$

mit Nullmittelwert- und Kovarianzfunktion  $k(\cdot, \cdot; \boldsymbol{\eta})$  beschrieben. Hierbei wurde die abkürzende Schreibweise  $\mathbf{x}_k^u := [\mathbf{x}_k^T, \mathbf{u}_k^T]^T$  eingeführt. Die Mittelwertfunktion ist null, da die Modellfehler unbekannt sind und das gesamte Vorwissen über die Dynamik bereits über das physikalische Modell erfasst wird. Des Weiteren ist eine geeignete Kovarianzfunktion  $k(\mathbf{x}, \mathbf{x}')$  für die vollständige Definition des GPs auszuwählen. In diesem Zusammenhang beschreibt die Kovarianzfunktion bzw. der Kernel die Eigenschaften der gesuchten Funktion in Bezug auf ihre Glattheit und den Grad ihrer Differenzierbarkeit. Außerdem kann über sie eine bestimmte Periodizität ausgedrückt werden [124]. Entsprechend lässt sich über den Kernel erweitertes Vorwissen im Vergleich zur rudimentären Mittelwertfunktion einbeziehen und die Wahl des Kernels ist ein Entwurfsparameter, der auf die jeweilige Aufgabe angepasst werden muss. Der gängigste Ansatz ist der sogenannte Squared Exponential (SE) Kernel, welcher hier beispielhaft vorgestellt wird. Der SE-Kernel hat die Form

$$k_{SE}(\mathbf{x}_k^u, \mathbf{x}_k^{u'}; \boldsymbol{\eta}) = \sigma_f^2 \exp(-(\mathbf{x}_k^u - \mathbf{x}_k^{u'})^T \mathbf{W}^{-1} (\mathbf{x}_k^u - \mathbf{x}_k^{u'})) \quad (3.117)$$

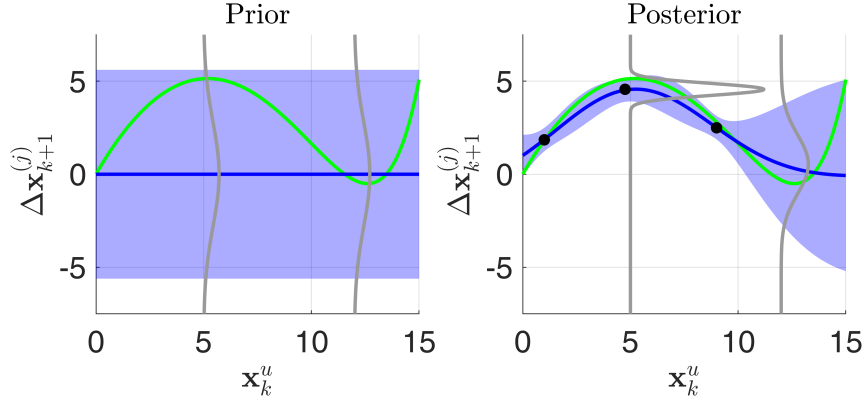
mit Gewichtungsmatrix  $\mathbf{W} = \text{diag}(l_1^2, \dots, l_{n_x}^2)$ , welche als Elemente die sogenannten Lengthscale-Parameter besitzt, sowie der Signalvarianz  $\sigma_f^2$  hat. Die parametrische Abhängigkeit ist gewollt, um dem GP eine gewisse Flexibilität einzuräumen und damit er sich auf unterschiedliche Rahmenbedingungen anpassen kann. Daher werden die folgenden Hyperparameter über  $\boldsymbol{\eta} := [l_1, \dots, l_{n_x}, \sigma_f]$  eingeführt, die für die vollständige Definition des GPs notwendig sind. Die Hyperparameter werden über die Minimierung der negativen logarithmischen Likelihood [124] nach

$$\boldsymbol{\eta}_j^* = \arg \min_{\boldsymbol{\eta}_j} \Delta \mathbf{X}_{k+1}^{(j,:)} \mathbf{K}_{\boldsymbol{\eta}_j}^{-1} \Delta \mathbf{X}_{k+1}^{(j,:)} + \log |\mathbf{K}_{\boldsymbol{\eta}_j}| \quad (3.118)$$

bestimmt. Dabei ist  $\Delta \mathbf{X}_{k+1}^{(j,:)}$  die  $j$ -te Zeile von  $\Delta \mathbf{X}_{k+1}$  und  $\mathbf{K}_{\boldsymbol{\eta}_j} \in \mathbb{R}^{n_d \times n_d}$  ist die symmetrische und positiv definite Gram-Matrix mit den Elementen

$$K_{\boldsymbol{\eta}_j}^{(r,c)} = k(\mathbf{X}_k^{U(r)}, \mathbf{X}_k^{U(c)}; \boldsymbol{\eta}_j), \quad r, c = 1, \dots, n_d, \quad (3.119)$$

mit  $\mathbf{X}_k^U := [\mathbf{X}_k^T, \mathbf{U}_k^T]^T$  und wobei der zusätzliche Index die benutzte Spalte angibt. Die Optimierung in Gleichung (3.118) stellt ein Mehrzielproblem dar, wobei die Problematik des Over- and Underfitting [10] adressiert wird und die Hyperparameter so bestimmt werden, dass die Messdaten bei einer möglichst geringen Modellkomple-



**Abbildung 3.29** Eindimensionale Visualisierung der Gauß-Prozess-Regression, links Prior- und rechts Posterior-Wahrscheinlichkeitsdichteverteilung. Die grüne Linie zeigt die wahre unbekannte Funktion des Modellfehlers (Ground Truth). Die blaue Linie repräsentiert den Mittelwert und die blaue Fläche die zweifache Standardabweichung des Gauß-Prozesses. Eine Auswertung dieser Größen findet auszugsweise an den Stellen  $x_k^u = 5$  und  $x_k^u = 12.5$  statt (graue Linien). Während der Prior noch uninformativ ist und damit eine hohe Unsicherheit aufweist, ist der Posterior an die Messpunkte (schwarze Punkte) angepasst und ist in der Lage den Ground Truth besser wiederzugeben. Außerdem erhält man über das Maß der Standardabweichung Informationen über die Unsicherheiten der betrachteten Größe.

xität gut wiedergegeben werden. Die Posterior-Wahrscheinlichkeitsdichteverteilung bzgl. der  $j$ -ten Dimension des Modellfehlervektors lautet dann

$$\begin{aligned} p(\Delta x_{k+1}^{(j)} \mid \Delta \mathbf{X}_{k+1}^{(j,:)}) &= \mathcal{N}(\Delta \mu_j(\mathbf{x}_k^u), \Delta \sigma_j^2(\mathbf{x}_k^u)), \\ \Delta \mu_j(\mathbf{x}_k^u) &= \mathbf{k}_\Delta(\mathbf{x}_k^u)^T \mathbf{K}_{\eta_j^*}^{-1} \Delta \mathbf{X}_{k+1}^{(j,:)^T}, \\ \Delta \sigma_j^2(\mathbf{x}_k^u) &= k(\mathbf{x}_k^u, \mathbf{x}_k^u; \boldsymbol{\eta}_j^*) - \mathbf{k}_\Delta(\mathbf{x}_k^u)^T \mathbf{K}_{\eta_j^*}^{-1} \mathbf{k}_\Delta(\mathbf{x}_k^u), \end{aligned} \quad (3.120)$$

mit  $\mathbf{k}_\Delta(\mathbf{x}_k^u) = [k(\mathbf{x}_k^u, \mathbf{X}_k^{U(1)}; \boldsymbol{\eta}_j^*), \dots, k(\mathbf{x}_k^u, \mathbf{X}_k^{U(n_d)}; \boldsymbol{\eta}_j^*)]^T$  als Vektor, welcher den Funktionseingang  $\mathbf{x}_k^u$  punktweise mit den Messdaten auswertet. Die Abbildung 3.29 visualisiert hierzu die grundlegenden Zusammenhänge an einem eindimensionalen Beispiel. Es ist gut erkennbar, dass sich die Verteilung über Gleichung (3.120) an die schwarzen Datenpunkte anpasst und die Unsicherheit in Form der Varianz in der Nähe der Datenpunkte im Vergleich zur Prior-Verteilung stark sinkt. Unter Verwendung der Standardannahme, dass die dimensionszugehörigen GPs unabhängig von einander sind, lautet das gesamte datengetriebene Modell damit formal

$$\begin{aligned} p(\Delta \mathbf{x}_{k+1} \mid \Delta \mathbf{X}_{k+1}) &= \mathcal{N}(\Delta \boldsymbol{\mu}(\mathbf{x}_k^u), \Delta \boldsymbol{\Sigma}(\mathbf{x}_k^u)), \\ \Delta \boldsymbol{\mu}(\mathbf{x}_k^u) &= [\Delta \mu_1(\mathbf{x}_k^u), \dots, \Delta \mu_{n_x}(\mathbf{x}_k^u)]^T, \\ \Delta \boldsymbol{\Sigma}(\mathbf{x}_k^u) &= \text{diag}([\Delta \sigma_1^2(\mathbf{x}_k^u), \dots, \Delta \sigma_{n_x}^2(\mathbf{x}_k^u)]). \end{aligned} \quad (3.121)$$

Entsprechend zu [125] wird für das aufgestellte GP-Modell eine Kurzschreibweise durch

$$\mathbf{f}_\Delta(\mathbf{x}_k, \mathbf{u}_k) := [\Delta \boldsymbol{\mu}(\mathbf{x}_k, \mathbf{u}_k), \Delta \boldsymbol{\Sigma}(\mathbf{x}_k, \mathbf{u}_k)], \quad (3.122)$$

mit der Abbildung  $\mathbf{f}_\Delta : \mathbb{R}^{n_x} \times \mathbb{R}^{n_u} \mapsto \mathbb{R}^{n_x} \times \mathbb{R}^{n_x \times n_x}$  eingeführt. Bei der Vernachlässigung der Kovarianzmatrix kann eine sogenannte naive Prädiktion des hybriden Modells nur mit Hilfe des Erwartungsvektors durchgeführt werden. Die Berechnung des Folgezustands ergibt sich demnach, entsprechend dem Ansatz aus Gleichung (3.113), mit

$$\mathbf{x}_{k+1} = \hat{\mathbf{f}}(\mathbf{x}_k, \mathbf{u}_k) + \Delta\boldsymbol{\mu}(\mathbf{x}_k, \mathbf{u}_k). \quad (3.123)$$

Dabei handelt es sich um eine klassische deterministische Prädiktion, welche die Unsicherheit über die Modellfehler durch eine zu geringe Datenlage vernachlässigt. Ein genaueres Bild des Systemverhaltens folgt aus der Berücksichtigung der Kovarianzmatrix, wobei eine probabilistische Zustandspropagation erforderlich wird. Eine zugehörige Vorschrift für eine Langzeitvorhersage des realen Systems wird im nächsten Abschnitt entwickelt und erläutert.

#### 3.5.2 Effiziente probabilistische Zustandspropagation mittels Unscented Transformation

Bis hierin wurde von einem eindeutigen deterministischen Zustandsvektor  $\mathbf{x}_k$  ausgegangen. Zudem wurde der Vektor der Modellfehler  $\Delta\mathbf{x}_k$  als Zufallsvektor definiert. Auf der Basis des Zusammenhangs des hybriden Ansatzes (3.113) geht hervor, dass es sich bei dem Folgezustand  $\mathbf{x}_{k+1}$  ebenfalls um einen Zufallsvektor handeln muss, da dieser direkt von dem Vektor der Modellfehler abhängt. Die Unsicherheit über die Modellfehler propagiert sich somit bei der Betrachtung mehrerer Zeitschritte bzw. bei einer Langzeitprädiktion für alle Zustandsvektoren fort. Da es sich bei der Dynamikgleichung (3.113) im Allgemeinen um einen nichtlinearen Zusammenhang handelt, sind die Zustandsverteilungen komplex und müssten aufwendig numerisch berechnet werden. Das erfordert viel Zeit und ist ineffizient.

Eine Möglichkeit für eine effiziente Propagation, welche in diesem Abschnitt vorgestellt wird, stellt das Moment Matching (MM) [126] in Kombination mit der Unscented Transformation (UT) [127] dar. Dabei werden die komplexen wahren Zustandsverteilungen mit einer Normalverteilung  $p(\mathbf{x}_k) \approx \mathcal{N}(\mathbf{m}_k, \mathbf{S}_k)$  approximiert, wobei der Erwartungsvektor  $\mathbf{m}_k$  und die Kovarianzmatrix  $\mathbf{S}_k$  durch die besonders effiziente UT berechnet werden. Die UT stellt in der Regelungstechnik ein gängiges Beobachterverfahren zur Zustandsschätzung dar.

Den Ausgangspunkt der Entwicklung stellt ein normalverteilter aktueller Zustand mit  $\mathbf{x}_k \sim \mathcal{N}(\mathbf{m}_k, \mathbf{S}_k)$  dar. Die Stellgröße  $\mathbf{u}_k$  bleibt weiterhin eindeutig, d. h. ohne Unsicherheiten, bekannt, da sie bei der Verwendung eines Steuerungsansatzes vorgegeben wird. Im Falle einer Regelung würde die Stellgröße vom Zustand abhängen, wodurch, wie bereits oben beschrieben, die Stellgröße ebenfalls als Zufallsvariable definiert werden müsste. Dieser Fall soll aber zunächst nicht betrachtet werden, sodass von einer deterministischen Stellgröße ausgegangen wird. Auf der Basis von Gleichung (3.113) wird zunächst das Gesetz der totalen Erwartung angewendet

$$\mathbf{m}_{k+1} = \mathbb{E}_{x_k}[\mathbb{E}_{\Delta x_{k+1}}[\mathbf{x}_{k+1}]] = \mathbb{E}_{x_k}[\hat{\mathbf{f}}] + \mathbb{E}_{x_k}[\Delta\boldsymbol{\mu}], \quad (3.124)$$

mit dem Erwartungswertoperator  $\mathbb{E}$ , um den Erwartungsvektor des nächsten Zu-

stands zu bestimmen. Dieser setzt sich additiv aus der Erwartung des physikalischen Modells und des Modellfehlers bzgl. der Unsicherheit des aktuellen Zustands zusammen. Die Approximation dieser Größen findet nach der Betrachtung der Zustandsvarianz statt. Die Zustandsvarianz wird entsprechend mit dem Gesetz der totalen Varianz über

$$\begin{aligned}\mathbf{S}_{k+1} &= \mathbb{E}_{x_k}[\mathbb{V}_{\Delta x_{k+1}}[\mathbf{x}_{k+1}]] + \mathbb{V}_{x_k}[\mathbb{E}_{\Delta x_{k+1}}[\mathbf{x}_{k+1}]] \\ &= \mathbb{E}_{x_k}[\Delta \Sigma] + \mathbb{V}_{x_k}[\hat{\mathbf{f}}] + \mathbb{V}_{x_k}[\Delta \mu] + \mathbb{C}_{x_k}[\hat{\mathbf{f}}, \Delta \mu] + \mathbb{C}_{x_k}[\hat{\mathbf{f}}, \Delta \mu]^T,\end{aligned}\quad (3.125)$$

mit Kovarianz-  $\mathbb{V}_{x_k}$  und Kreuzkovarianzoperator  $\mathbb{C}_{x_k}$ , bestimmt und setzt sich damit aus verschiedenen Summanden zusammen. Die Berechnung der verbliebenen Operatoren in Gleichung (3.124) und (3.125) erfolgt durch die Anwendung der numerisch effizienten UT. Diese sieht zunächst eine Aufstellung von  $2n_x + 1$  sogenannten Sigma-Punkten vor, welche sich aus der aktuellen Zustandsverteilung  $\mathbf{x}_k \sim \mathcal{N}(\mathbf{m}_k, \mathbf{S}_k)$  mit

$$\begin{aligned}\mathbf{x}_{s,k}^{(i)} &= \mathbf{m}_k + (\sqrt{(\kappa + n_x)\mathbf{S}_k})_{(i,:)}^T, \quad i = 1, \dots, n_x, \\ \mathbf{x}_{s,k}^{(i)} &= \mathbf{m}_k - (\sqrt{(\kappa + n_x)\mathbf{S}_k})_{(i,:)}^T, \quad i = n_x + 1, \dots, 2n_x, \\ \mathbf{x}_{s,k}^{(2n_x+1)} &= \mathbf{m}_k,\end{aligned}\quad (3.126)$$

ergeben. Dabei ist  $(\sqrt{(\kappa + n_x)\mathbf{S}_k})_{(i,:)}^T$  die  $i$ -te Zeile der Matrixwurzel und  $\kappa \in \mathbb{R}$  ein a-priori festzulegender Entwurfsparemeter mit der Bedingung  $\kappa + n_x \neq 0$ . Im nächsten Schritt werden die beiden Modellteile des hybriden Ansatzes bei den Sigma-Punkten ausgewertet:

$$\begin{aligned}\mathbf{x}_{s,k+1}^{(i)} &= \hat{\mathbf{f}}(\mathbf{x}_{s,k}^{(i)}, \mathbf{u}_k), \\ [\Delta \mu_s^{(i)}, \Delta \Sigma_s^{(i)}] &= \mathbf{f}_\Delta(\mathbf{x}_{s,k}^{(i)}, \mathbf{u}_k), \quad i = 1, \dots, 2n_x + 1.\end{aligned}\quad (3.127)$$

Anschließend werden die transformierten Sigma-Punkte bzw. die Funktionsauswertungen verwendet, um die gesuchten Momente numerisch zu approximieren,

$$\begin{aligned}\mathbb{E}_{x_k}[\hat{\mathbf{f}}] &\approx \sum_{i=1}^{2n_x+1} w_s^{(i)} \mathbf{x}_{s,k+1}^{(i)} =: \mathbf{m}_{\hat{\mathbf{f}}}, \\ \mathbb{V}_{x_k}[\hat{\mathbf{f}}] &\approx \sum_{i=1}^{2n_x+1} w_s^{(i)} (\mathbf{x}_{s,k+1}^{(i)} - \mathbf{m}_{\hat{\mathbf{f}}})(\mathbf{x}_{s,k+1}^{(i)} - \mathbf{m}_{\hat{\mathbf{f}}})^T, \\ \mathbb{E}_{x_k}[\Delta \mu] &\approx \sum_{i=1}^{2n_x+1} w_s^{(i)} \Delta \mu_s^{(i)} =: \mathbf{m}_{\Delta \mu}, \\ \mathbb{E}_{x_k}[\Delta \Sigma] &\approx \sum_{i=1}^{2n_x+1} w_s^{(i)} \Delta \Sigma_s^{(i)}, \\ \mathbb{V}_{x_k}[\Delta \mu] &\approx \sum_{i=1}^{2n_x+1} w_s^{(i)} (\Delta \mu_s^{(i)} - \mathbf{m}_{\Delta \mu})(\Delta \mu_s^{(i)} - \mathbf{m}_{\Delta \mu})^T, \\ \mathbb{C}_{x_k}[\hat{\mathbf{f}}, \Delta \mu] &\approx \sum_{i=1}^{2n_x+1} w_s^{(i)} (\mathbf{x}_{s,k+1}^{(i)} - \mathbf{m}_{\hat{\mathbf{f}}})(\Delta \mu_s^{(i)} - \mathbf{m}_{\Delta \mu})^T.\end{aligned}\quad (3.128)$$

Dabei werden die Gewichtungen

$$w_s^{(i)} = \begin{cases} \frac{1}{2(n_x + \kappa)}, & i = 1, \dots, 2n_x, \\ \frac{\kappa}{n_x + \kappa}, & i = 2n_x + 1, \end{cases} \quad (3.129)$$

benutzt, deren Herleitung eine möglichst hohe Approximationsgüte zum Ziel hat [127]. Zu beachten ist, dass trotz der diagonalen Struktur der GP-Kovarianzmatrix (3.121) bzw. der Unabhängigkeitsannahme, die Dimensionen des Nachfolgezustands für eine unsichere Eingangsverteilung kovariieren können. Die Gleichungen aus (3.124)-(3.128) werden zusammen als eine mathematische Funktion definiert, um eine Kurzschreibweise einzuführen. Diese Funktion wird mit

$$\tilde{\mathbf{f}}(\mathbf{m}_k, \mathbf{S}_k, \mathbf{u}_k) := [\mathbf{m}_{k+1}, \mathbf{S}_{k+1}] \quad (3.130)$$

beschrieben, wobei die aktuelle Zustandsverteilung zusammen mit der Stellgröße auf die nächste Zustandsverteilung abgebildet werden. Eine probabilistische Langzeitprädiktion ergibt sich für eine bekannte Steuerungsabfolge durch eine mehrfache Ausführung dieser Funktion, ähnlich zu einer diskreten Dynamikfunktion im deterministischen Fall.

Für einige Systeme ist es nicht notwendig, alle Dynamikgleichungen zu korrigieren. Das Systemverhalten setzt sich bspw. aus verschiedenen physikalischen Effekten zusammen und unter Umständen ist ein Teil dieser Effekte klar und eindeutig bestimmbar, sodass die zugehörigen Gleichungen keinen Modellfehler enthalten. Insbesondere bei mechanischen Systemen ist häufig ein Integratorverhalten bekannt, wobei eine Zustandsgröße lediglich aufintegriert wird. Auch in diesem Fall ist keine Korrektur vorzunehmen. Für den Rahmen der entwickelten Gleichungen kann dies über den Modellansatz

$$\mathbf{x}_{k+1} = \hat{\mathbf{f}}(\mathbf{x}_k, \mathbf{u}_k) + \mathbf{B}\Delta\mathbf{x}_{k+1}(\mathbf{x}_k, \mathbf{u}_k) \quad (3.131)$$

abgebildet werden, wobei eine Kopplungsmatrix  $\mathbf{B} \in \mathbb{R}^{n_x \times n_b}$  mit  $n_b \leq n_x$  verwendet wird. Die Kopplungsmatrix legt dann entsprechend fest, welche Gleichungen einer Korrektur bedürfen. In diesem Zusammenhang hat  $\Delta\mathbf{x}_{k+1}(\mathbf{x}_k, \mathbf{u}_k)$  eine niedrigere Dimension als der Zustandsvektor. Vereinfachend wird jedoch keine neue Größe definiert, sondern die bestehende Notation beibehalten. Die Dimension erschließt sich dann aus dem Kontext. In diesem Szenario sind lediglich die Gleichungen (3.124) und (3.125) durch

$$\begin{aligned} \mathbf{m}_{k+1} &= \mathbb{E}_{x_k}[\hat{\mathbf{f}}] + \mathbf{B}\mathbb{E}_{x_k}[\Delta\boldsymbol{\mu}], \\ \mathbf{S}_{k+1} &= \mathbf{B}\mathbb{E}_{x_k}[\Delta\boldsymbol{\Sigma}]\mathbf{B}^T + \mathbb{V}_{x_k}[\hat{\mathbf{f}}] + \mathbf{B}\mathbb{V}_{x_k}[\Delta\boldsymbol{\mu}]\mathbf{B}^T \\ &\quad + \mathbb{C}_{x_k}[\hat{\mathbf{f}}, \Delta\boldsymbol{\mu}]\mathbf{B}^T + \mathbf{B}\mathbb{C}_{x_k}[\hat{\mathbf{f}}, \Delta\boldsymbol{\mu}]^T \end{aligned} \quad (3.132)$$

anzupassen, welche sich durch eine lineare Transformation einer Normalverteilung ergeben. Ein weiterer Spezialfall liegt vor, wenn es keinerlei Vorwissen über die Dynamik des betrachteten Systems gibt. Dann gilt  $\hat{\mathbf{f}} = \mathbf{0}$  und der Modellansatz

### 3 Modellbildung

vereinfacht sich zu

$$\mathbf{x}_{k+1} = \Delta \mathbf{x}_{k+1}(\mathbf{x}_k, \mathbf{u}_k), \quad (3.133)$$

wobei lediglich der datengetriebene Teil bestehen bleibt. Dementsprechend ergibt sich für die Zustandspropagation

$$\begin{aligned} \mathbf{m}_{k+1} &= \mathbb{E}_{x_k}[\Delta \boldsymbol{\mu}], \\ \mathbf{S}_{k+1} &= \mathbb{E}_{x_k}[\Delta \boldsymbol{\Sigma}] + \mathbb{V}_{x_k}[\Delta \boldsymbol{\mu}]. \end{aligned} \quad (3.134)$$

Die hybride Modellierung für partiell bekannte dynamische Systeme mittels der Gauß-Prozess-Regression fasst den Modellfehler als Zufallsvariable auf und korrigiert ihn datengetrieben. Eine numerisch effiziente Simulation ist in Form einer Zustandspropagation durch die Unscented Transformation gegeben. Der Rechenaufwand ist dabei im Vergleich zu einer gängigen deterministischen Simulation höher, jedoch können aus der probabilistischen Simulation zusätzliche Informationen über die Unsicherheit des Zustands, welche durch die unbekannten Modellfehler hervorgerufen wird, gewonnen werden. Auf diese Weise lässt sich das zeitliche Verhalten am realen System besser abschätzen und geeignete Sicherheitsmaßnahmen für einen sicheren und stabilen Betrieb ableiten.

## 4 Steuerungs- und Regelungsentwurf

Es existieren zwei verschiedene Möglichkeiten, um eine hybride Steuerung oder Regelung zu erhalten. Zum einen kann die Grundlage ein zunächst mit Hilfe von ML-Ansätzen entwickeltes Modell des betrachteten Systems sein, wie sie beispielsweise in Kapitel 3 vorgestellt werden. Wenn dieses hybride Modell dann für den Steuerungs- oder Regelungsentwurf verwendet wird, müssen zumeist die zusätzlichen neuen Eigenschaften aus dem hybriden Ansatz ebenfalls berücksichtigt werden. Diese Herangehensweise wird auch als indirekte Methode [128] bezeichnet und ein Beispiel hierfür ist die hybride Optimalsteuerung aus Unterabschnitt 4.4.1. Hier wird ein probabilistisches Modell betrachtet, welches Modellfehler in der Inbetriebnahme lernen soll. Dies hat aber auch Konsequenzen für die Optimalsteuerung, in der nun auch der Umgang mit der probabilistischen Sichtweise berücksichtigt werden muss. Eine andere Herangehensweise ist es, einen Regelungsentwurf direkt so zu verändern, dass er hybrid ist, was auch als direkte Methode bezeichnet wird [128]. Dies geschieht beispielsweise in der hybriden Zustandslinearisierung in Unterabschnitt 4.4.2. Hier werden die Gleichungen der Zustandslinearisierung direkt so beeinflusst, dass sich eine hybride Methode ergibt, welche die Kompensationsfunktionen anhand von Messdaten anpasst. In beiden Fällen muss in den Steuerungs- und Regelungsansätzen untersucht werden, wie sich die ML-Ansätze auswirken und welche Vor- oder auch Nachteile sich ergeben. Eine hybride Modellierung und hybrider Steuerungs- bzw. Regelungsentwurf bedeutet fast immer einen höheren Entwicklungs- oder auch Rechenaufwand. Daher sollte vorab immer die Notwendigkeit dieser Ansätze überprüft werden und Kosten gegen Nutzen abgeschätzt werden.

### 4.1 Riccati-Regelungen mittels des Koopman-Operators

Der Koopman-Operator bietet das Potenzial, lineare Entwurfsmethoden für die Regelung von nichtlinearen Streckendynamiken zu nutzen. In [27] wird erstmals der Entwurf eines linearen Riccati-Reglers für nichtlineare Dynamiken im höherdimensionalen Koopman-Raum präsentiert und als *Koopman Operator Optimal Control* bezeichnet. Im Folgenden werden die Grundidee anhand eines Einführungsbeispiels sowie mögliche Erweiterungen vorgestellt. Anschließend wird der Entwurf eines Riccati-Reglers beispielhaft am EDMD-Modell des Schlagmechanismus des Golfroboters demonstriert.

#### 4.1.1 Illustration der Idee

Die Idee eines Koopman-basierten linearen Riccati-Reglers wird anhand des Einführungsbeispiels (3.8) illustriert. Ergänzt um einen Steuerungseingang und so modifi-

ziert, dass der Zustand  $x_2$  sich instabil verhält, ergibt sich für die Differentialgleichungen

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} \mu x_1 \\ \lambda(x_2 - x_1^2) + u \end{bmatrix} \quad \text{mit} \quad \lambda = 1, \quad \mu = -0,05. \quad (4.1)$$

Die quadratische Kostenfunktion für den Entwurf des Riccati-Reglers wird als

$$J = \int_0^\infty \mathbf{x}^\top(t) \mathbf{Q} \mathbf{x}(t) + R u^2(t) dt \quad \text{mit} \quad \mathbf{Q} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \quad R = 1 \quad (4.2)$$

angenommen. Das für die Berechnung des Regelungsgesetzes erforderliche lineare Streckenmodell wird klassischerweise mittels einer exakten Zustandslinearisierung oder einer Linearisierung der Dynamik um einen oder mehrere Arbeitspunkte bestimmt. Auf der einen Seite wird für die Zustandslinearisierung  $u = v + \lambda x_1^2$  gewählt, sodass sich das resultierende lineare System

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} \mu & 0 \\ 0 & \lambda \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} v \quad (4.3)$$

mit dem neuen Eingang  $v$  und dem nichtlinearen Regelungsgesetz

$$u = v + \lambda x_1^2 = -\mathbf{k}^\top \mathbf{x} + \lambda x_1^2 \quad \text{mit} \quad \mathbf{k}^\top = [0, 2,414] \quad (4.4)$$

ergibt. Auf der anderen Seite führt eine Linearisierung um die Ruhelage  $\mathbf{x}^* = \mathbf{0}$  zum linearen Modell

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} \mu & 0 \\ 0 & \lambda \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u \quad (4.5)$$

mit dem resultierenden linearen Regelungsgesetz

$$u = -\mathbf{k}^\top \mathbf{x} \quad \text{mit} \quad \mathbf{k}^\top = [0, 2,424]. \quad (4.6)$$

Im Gegensatz zu den beiden klassischen Ansätzen ermöglicht die Koopman-Beschreibung

$$\begin{bmatrix} \dot{g}_1 \\ \dot{g}_2 \\ \dot{g}_3 \end{bmatrix} = \begin{bmatrix} \mu & 0 & 0 \\ 0 & \lambda & -\lambda \\ 0 & 0 & 2\mu \end{bmatrix} \begin{bmatrix} g_1 \\ g_2 \\ g_3 \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} u \quad (4.7)$$

der Originaldynamik, vgl. Abschnitt 3.1, eine unmittelbare, d. h. linearisierungsfreie Anwendung des Riccati-Entwurfs. Die Kostenfunktion

$$\tilde{J} = \int_0^\infty \mathbf{g}^\top(\mathbf{x}(t)) \tilde{\mathbf{Q}} \mathbf{g}(\mathbf{x}(t)) + R u^2(t) dt \quad \text{mit} \quad \tilde{\mathbf{Q}} = \begin{bmatrix} \mathbf{Q} & 0 \\ 0 & 0 \end{bmatrix} \quad (4.8)$$

wird identisch zu Gleichung (4.2) gewählt<sup>1</sup> und liefert das nichtlineare Regelgesetz

$$u = -\tilde{\mathbf{k}}^\top \mathbf{g}(\mathbf{x}) = -[\tilde{k}_1, \tilde{k}_2] \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} - \tilde{k}_3 x_1^2 \quad \text{mit} \quad \tilde{\mathbf{k}}^\top = [0, 2,414, -1,594], \quad (4.9)$$

<sup>1</sup>Für das hier betrachtete Beispiel genügt es, dass  $\tilde{\mathbf{Q}}$  positiv semidefinit ist, weil zusätzlich das betrachtete System über die Ausgangsgleichung  $\mathbf{y} = \tilde{\mathbf{Q}}_0 \mathbf{x}$  mit  $\tilde{\mathbf{Q}}_0^\top \tilde{\mathbf{Q}}_0$  beobachtbar ist, vgl. [24].



das eine ähnliche Struktur wie Gleichung (4.4) aufweist. Bemerkenswert ist, dass die Reglerparameter  $[\tilde{k}_1, \tilde{k}_2]$  mit  $\mathbf{k}^\top$  übereinstimmen und der einzige Unterschied zwischen der Koopman-basierten Regelung und der exakten Zustandslinearisierung im Faktor  $-\tilde{k}_3 \neq \lambda$  besteht.

Die Abbildung 4.1 zeigt die simulative Untersuchung des Koopman-basierten Riccati-Reglers. Bezüglich der Minimierung der Kostenfunktion (4.2) ist die auf dem Koopman-Modell basierende Zustandsrückführung den beiden klassischen Verfahren überlegen. Dass der Zustand  $x_2$  der auf einer Zustandslinearisierung basierenden Regelung in der Abbildung 4.1(a) schneller abklingt, lässt sich dadurch erklären, dass die Kostenfunktion (4.2) für diesen Fall fälschlicherweise den neuen Eingang  $v$  anstatt des wahren Eingangs  $u$  gewichtet und dadurch den Stellgrößenverbrauch unterschätzt. In der Abbildung 4.1(b) sind die geregelten Trajektorien für unterschiedliche Anfangsauslenkungen dargestellt. Beim Koopman-basierten Ansatz scheint die charakteristische langsame Mannigfaltigkeit, vgl. Abbildung 3.3, die Dynamik des geregelten Systems maßgeblich zu bestimmen, weil sich sämtliche Trajektorien zunächst asymptotisch dieser annähern, bevor sie schließlich in den Ursprung streben, vgl. Abbildung 3.3. Der für die Koopman-basierte Regelung erheblich reduzierte Stellgrößenverbrauch, der in der Abbildung 4.1(a) zu erkennen ist, resultiert aus der geschickten Nutzung dieses Phänomens.

Das betrachtete Einführungsbeispiel stellt aufgrund des Koopman-invarianten Unterraums einen Spezialfall dar. Der Beitrag [27] weist darauf hin, dass das Entwurfsvorgehen an seine Grenzen stößt, sobald einzelne Zustände in  $\mathbf{g}$  nicht steuerbar seien oder es nicht möglich sei, eine endlichdimensionale Koopman-invariante Systembeschreibung zu bestimmen. Gleichwohl wurde das Verfahren in nachfolgenden Arbeiten erfolgreich auch auf numerisch approximierte EDMD-Modelle, vgl. Unterabschnitt 3.1.4 angewendet. Zahlreiche Veröffentlichungen demonstrieren die Effektivität Koopman-basierter Riccati-Regler anhand unterschiedlicher Beispielsysteme<sup>2</sup>.

Der Beitrag [129] präsentiert analog zur oben beschriebenen Strategie einen Ansatz zur Regelung mittels Koopman-Eigenfunktionen, wobei die Kostenfunktion

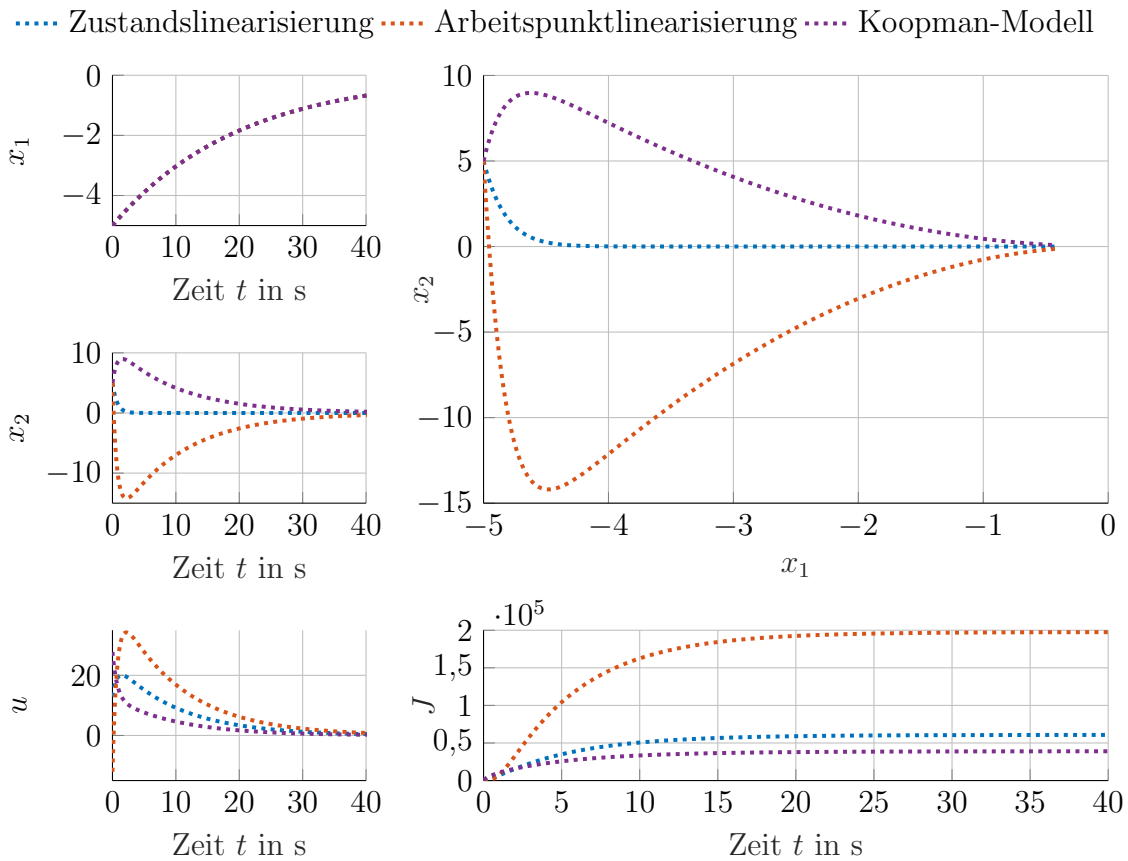
$$\tilde{J} = \int_0^\infty \boldsymbol{\varphi}^\top(\mathbf{x}(t)) \tilde{\mathbf{Q}} \boldsymbol{\varphi}(\mathbf{x}(t)) + Ru^2(t) dt \quad \text{mit} \quad \tilde{\mathbf{Q}} = \begin{bmatrix} \mathbf{Q} & 0 \\ 0 & 0 \end{bmatrix} \quad (4.10)$$

so gewählt wird, dass  $\boldsymbol{\varphi}^\top \tilde{\mathbf{Q}} \boldsymbol{\varphi} \approx \mathbf{x}^\top \mathbf{Q} \mathbf{x}$ . Dieser Ansatz resultiert im nichtlinearen Regelungsgesetz

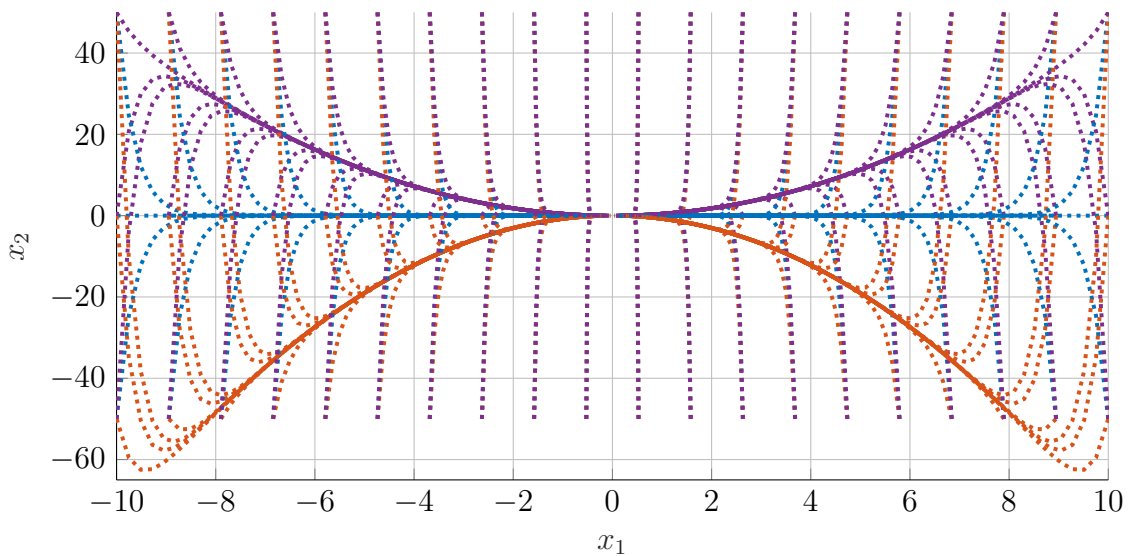
$$\mathbf{u} = -\tilde{\mathbf{k}}^\top \boldsymbol{\varphi}(\mathbf{x}). \quad (4.11)$$

Die Regelgüte wird anhand einer Energieregulation für Hamiltonsche Systeme demonstriert, bei denen die Koopman-Eigenfunktion die im System gespeicherte Energie mit dem Koopman-Eigenwert  $\lambda = 0$  ist. Darüber hinaus finden sich in der Literatur weitere Ansätze zur Regelung mittels Koopman-Eigenfunktionen. In [138]

<sup>2</sup>In Form von Simulationsstudien wird die Regelung für ein Pendel auf dem Wagen [117], den Duffing-Oszillator und den Van-der-Pol-Oszillator [130], einen Boost-Konverter mit aktiver Last [131], das FitzHugh-Nagumo-Modell und die Lorenz-Gleichungen [132], die Dynamik starrer Körper in Dualquaternionendarstellung [133], einen Spurhalteassistenten [134], einen MEMS-Drehratensensor [135] und weitere akademische Beispiele [136] beschrieben. In Form von Experimenten demonstrieren [117] und [137] den Regelungsentwurf für Spielzeug- und Industrieroboter, [45] für einen schwimmenden Roboterfisch mit aktuierter Schwanzflosse.



(a) Beispielhafte simulierte Regelung einer Anfangsauslenkung. Gemäß der Kostenfunktion  $J$ , vgl. (4.2), ist die Zustandsrückführung (4.9), die auf dem Koopman-Modell (4.7) basiert, den Regelungsgesetzen (4.4) und (4.6), die auf einer Zustands- bzw. Arbeitspunktlinearisierung basieren, überlegen. In Anlehnung an [27], [129].



(b) Vergleich der unterschiedlichen Regelungsgesetze für unterschiedliche Anfangsauslenkungen.

**Abbildung 4.1** Veranschaulichung der Koopman-basierten Riccati-Regelung im Vergleich zu einer Zustandslinearisierung bzw. einer Linearisierung um einen Arbeitspunkt.

wird beispielsweise das Regelungsproblem für einen endlichen Zeithorizont formuliert. Der Beitrag [139] zeigt die Anwendung eines Riccati-Reglers mit Koopman-Eigenfunktionen für den Duffing- und den Van-der-Pol-Oszillator.

### 4.1.2 Beispielhafte Anwendung

Das Verfahren wurde beispielhaft anhand der Schlagregelung des Golfroboters, vgl. Abbildung 6.1, simulativ untersucht. Für den zeitinvarianten Riccati-Regler wurde ein zeitkontinuierliches EDMD-Modell mit  $N = 4$  Observablen (3.53) betrachtet, das anhand simulativ erzeugter Messdaten aus dem Modell (6.11) trainiert wurde. Mit der Kostenfunktion

$$\tilde{J} = \int_0^\infty \Psi^\top(\mathbf{x}(t)) \tilde{\mathbf{Q}} \Psi(\mathbf{x}(t)) + R u^2(t) \, dt \quad (4.12a)$$

$$\text{mit } \tilde{\mathbf{Q}} = \text{diag}(5, 1, 1 \cdot 10^{-12}, 1 \cdot 10^{-12}), \quad R = 1, \quad (4.12b)$$

die analog zu den Gleichungen (6.16)-(6.17) gewählt wurde, ergibt sich die Zustandsrückführung

$$u_c = \tilde{\mathbf{k}}^\top (\Psi^* - \Psi(\mathbf{x})) \quad \text{mit } \tilde{\mathbf{k}}^\top \in \mathbb{R}^{1 \times N}. \quad (4.13)$$

Mit den Führungsgrößen  $\tilde{\mathbf{w}} = \Psi(\mathbf{x}^*)$  ergeben sich die Sollverläufe

$$\Psi^* = \mathbf{F}_\Psi \tilde{\mathbf{w}}, \quad u^* = \tilde{\mathbf{f}}_u \tilde{\mathbf{w}} \quad (4.14)$$

mit den Vorsteuerungsmatrizen

$$\mathbf{F}_\Psi = \mathbf{I}_N, \quad \tilde{\mathbf{f}}_u = -\mathbf{b}^+ \mathbf{K} \quad \text{mit } \tilde{\mathbf{f}}_u \in \mathbb{R}^{1 \times N} \quad (4.15)$$

in Anlehnung an die Gleichungen (6.18)-(6.19) und die studentische Arbeit [140]. Das resultierende Regelgesetz lautet

$$u = u^* + u_c. \quad (4.16)$$

In der Abbildung 4.5 ist die resultierende Regelgüte (6.16) des EDMD-basierten Riccati-Reglers für eine beispielhafte Solltrajektorie mit einem Ausholwinkel von  $\varphi_1 = 120^\circ$ , einer Ausholdauer von  $T_a = T_r = 1 \text{ s}$  und einer Schlaggeschwindigkeit von  $\|\mathbf{v}_s\| = 3 \text{ m s}^{-1}$  dargestellt<sup>3</sup>. Es ist zu erkennen, dass die Koopman-basierte linear-quadratische (LQ-)Regelung eine deutlich höhere Regelgüte als die bisher verwendete rein physikalisch motivierte LQ-Regelung mit Gain-Scheduling aufweist. Die Abbildung 4.5(b) visualisiert die resultierende Regelgüte in Abhängigkeit der frei wählbaren Parameter der Solltrajektorie, vgl. Tabelle 6.1. Es ist zu erkennen, dass die EDMD-basierte LQ-Regelung die Regelungsaufgabe für alle wählbaren Werte sehr gut erfüllt, jedoch in besonderem Maße im Bereich kleiner Schlaggeschwindigkeiten. Eine mögliche Erklärung hierfür liegt darin, dass die nichtlinearen (Haft-) Reibungseffekte sich vor allem bei kleinen Geschwindigkeiten stark auf die Dynamik auswirken. Neben der hohen Regelgüte ist zusätzlich der stark verringerte Aufwand

<sup>3</sup>Diese Wahl der Parameter führt dazu, dass die nichtlinearen Anteile der Dynamik des Schlagmechanismus des Golfroboters hinreichend angeregt werden.

bei der Modellbildung durch das hybride Vorgehen bei diesem Ansatz hervorzuheben.

EDMD-Modelle lassen sich geradlinig für den Entwurf von **Riccati-Regelungen** verwenden. Durch die systematische Ausnutzung der Linearitätseigenschaft können somit **stabile nichtlineare Regelungsgesetze mithilfe eines linearen Entwurfsverfahrens** bestimmt werden. Für einen erfolgreichen Entwurf ist es entscheidend, dass die Kostenfunktion hierbei geschickt gewählt wird.

## 4.2 Modellprädiktive Regelung mittels des Koopman-Operators

Modellprädiktive Regelungsansätze (MPC) optimieren online anhand eines internen Streckenmodells den Ausgangsgrößenverlauf mittels des Stellgrößenverlaufs. Der optimierte Stellgrößenverlauf wird dann für die Regelung des betrachteten Systems verwendet. Die grundlegende Idee der MPC ist in der Abbildung 4.2 dargestellt. Ausgehend vom Zeitpunkt  $k - 1$  mit bekannter Stellgröße  $\mathbf{u}_{k-1}$  variiert die Optimierung den zukünftigen Stellgrößenverlauf  $\mathbf{u}_{k+i}$  ab dem Zeitpunkt  $k$  für eine endliche Zahl  $i = 0, \dots, n_c - 1$  von Stellgrößenschritten so, dass eine vorgegebene Kostenfunktion, beispielsweise

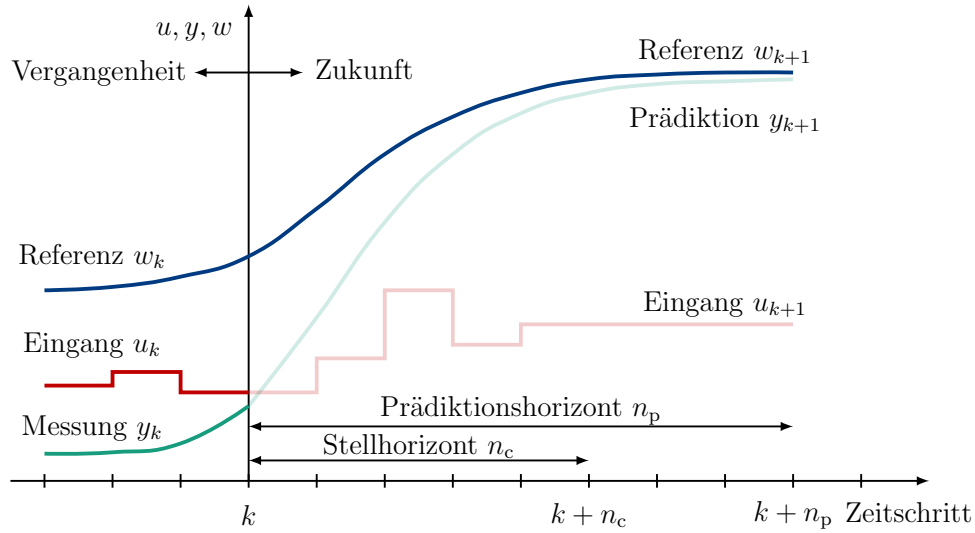
$$J = \sum_{i=1}^{n_p} \|\mathbf{Q}(\mathbf{y}_{k+i} - \mathbf{w}_{k+i})\|^2 + r \sum_{i=1}^{n_c} \|\mathbf{u}_{k+i-1}\|^2, \quad (4.17)$$

mit der positiv definiten Matrix  $\mathbf{Q} \in \mathbb{R}^{q \times q}$  und dem Faktor  $r \in \mathbb{R}$  über  $n_p$  Zeitschritte minimal wird. Die Werte  $n_p, n_c \in \mathbb{N}$  mit  $n_p \geq n_c$  werden als Prädiktions- bzw. Stellhorizont bezeichnet. Für den Zeitbereich  $i \geq n_c$  werden alle Stellgrößen auf  $\mathbf{u}_{k+n_c-1}$  gehalten, vgl. Abbildung 4.2. Nach der Optimierung wird nur der erste Wert der Stellgrößenfolge auf die reale Regelstrecke angewendet. Danach wird der Prädiktions- und Optimierungsprozess um einen Zeitschritt in die Zukunft verschoben, was als gleitender Horizont bezeichnet wird [21].

Abhängig vom verwendeten Streckenmodell gibt es unterschiedliche MPC-Ansätze. Lineare Streckenmodelle erlauben eine einfache Implementierung und Berechnung der MPC und ermöglichen – im Vergleich zu zeitinvarianten Regelungsansätzen – die Berücksichtigung von Stell-, Ausgangs- oder Zustandsgrößenbeschränkungen. Nichtlineare Streckenmodelle hingegen erschweren die Lösung des Optimierungsproblems. Dies resultiert aus der komplizierten Abhängigkeit der Kostenfunktion  $J$  von der Stellgrößenfolge  $\mathbf{u}_{k+i}$ , was einen erhöhten Rechenaufwand verursacht und dazu führt, dass die Auswertung der Kostenfunktion nicht konvex ist, vgl. [141]. Daher werden nichtlineare MPC in der Praxis gewöhnlich für eher langsame Systeme verwendet.<sup>4</sup>

Für den MPC-Entwurf lassen sich auch datengetriebene Streckenmodelle, beispielsweise DMD-Modelle [143], SINDy-Modelle [144] oder neuronale Netze [145] verwenden. Außerdem finden sich in der Literatur Ansätze für den direkten datengetriebenen Entwurf einer MPC, der unmittelbar auf Messdaten basiert, beispielsweise

<sup>4</sup>Beispielsweise für die Regelung eines Trinkwasserversorgungssystems [142].



**Abbildung 4.2** Grundlegende Idee der modellprädiktiven Regelung. In Anlehnung an [21].

in [146].

Die Nutzung von Koopman-Operator-basierten Streckenmodellen für die MPC erscheint vielversprechend, weil sie die Anwendung der Algorithmen der linearen MPC für nichtlineare Streckendynamiken ermöglichen. Im Folgenden wird die Berechnung der Koopman-Operator-basierten MPC vorgestellt und die Anwendung am Schlagmechanismus des Golfroboters demonstriert.

## Berechnung der MPC basierend auf EDMD-Modellen

Die Autor\*innen in [47] zeigen, dass EDMD-Modelle sich aufgrund der linearen Struktur geradlinig für den Entwurf linearer modellprädiktiver Regelungen für nichtlineare Streckendynamiken nutzen lassen. In Anlehnung an die klassische lineare MPC, vgl. [21], lässt sich für ein EDMD-Modell

$$\Psi(x_{k+1}) = K_t \Psi(x_k) + B_t u_k, \quad (4.18a)$$

$$y_k = C x_k = C P \Psi_k, \quad (4.18b)$$

vgl. Abschnitt 3.1, eine Vorschrift für den zeitlichen Verlauf der Observablen

$$\Psi_{k+1} = K_t \Psi_k + B_t u_k, \quad (4.19a)$$

$$\Psi_{k+2} = K_t \Psi_{k+1} + B_t u_{k+1} = K_t^2 \Psi_k + K_t B_t u_k + B_t u_{k+1}, \quad (4.19b)$$

$\vdots$

$$\Psi_{k+i} = K_t \Psi_{k+i-1} + B_t u_{k+i-1} = K_t^i \Psi_k + \sum_{j=1}^i K_t^{i-j} B_t u_{k+j-1} \quad (4.19c)$$

und des Systemausgangs

$$y_{k+1} = C x_{k+1} = C P \Psi_{k+1} = C P K_t \Psi_k + C P B_t u_k, \quad (4.20a)$$

$$y_{k+2} = C x_{k+2} = C P \Psi_{k+2} = C P K_t^2 \Psi_k + C P K_t B_t u_k + C P B_t u_{k+1}, \quad (4.20b)$$

$$\begin{aligned} & \vdots \\ \mathbf{y}_{k+i} &= \mathbf{C}\mathbf{x}_{k+i} = \mathbf{C}\mathbf{P}\Psi_{k+i} = \mathbf{C}\mathbf{P}\mathbf{K}_t^i \mathbf{x}_k + \sum_{j=1}^i \mathbf{C}\mathbf{P}\mathbf{K}_t^{i-j} \mathbf{B}_t \mathbf{u}_{k+j-1} \end{aligned} \quad (4.20c)$$

bestimmen. In Matrix-Vektor-Form mit

$$\bar{\mathbf{u}}_k = \begin{bmatrix} \mathbf{u}_k \\ \mathbf{u}_{k+1} \\ \vdots \\ \mathbf{u}_{k+n_c-1} \end{bmatrix} \in \mathbb{R}^{p \cdot n_c}, \quad \bar{\mathbf{y}}_{k+1} = \begin{bmatrix} \mathbf{y}_{k+1} \\ \mathbf{y}_{k+2} \\ \vdots \\ \mathbf{y}_{k+n_p} \end{bmatrix} \in \mathbb{R}^{q \cdot n_p}, \quad (4.21a)$$

$$\tilde{\mathbf{F}} = \begin{bmatrix} \mathbf{C}\mathbf{P}\mathbf{K}_t \\ \mathbf{C}\mathbf{P}\mathbf{K}_t^2 \\ \vdots \\ \mathbf{C}\mathbf{P}\mathbf{K}_t^{n_p} \end{bmatrix} \in \mathbb{R}^{q \cdot n_p \times N}, \quad (4.21b)$$

$$\tilde{\mathbf{H}} = \begin{bmatrix} \mathbf{C}\mathbf{P}\mathbf{B}_t & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{C}\mathbf{P}\mathbf{K}_t \mathbf{B}_t & \mathbf{C}\mathbf{P}\mathbf{B}_t & \cdots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{C}\mathbf{P}\mathbf{K}_t^{n_c-1} \mathbf{B}_t & \mathbf{C}\mathbf{P}\mathbf{K}_t^{n_c-2} \mathbf{B}_t & \cdots & \mathbf{C}\mathbf{P}\mathbf{B}_t \\ \mathbf{C}\mathbf{P}\mathbf{K}_t^{n_c} \mathbf{B}_t & \mathbf{C}\mathbf{P}\mathbf{K}_t^{n_c-1} \mathbf{B}_t & \cdots & \mathbf{C}\mathbf{P}\mathbf{K}_t \mathbf{B}_t \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{C}\mathbf{P}\mathbf{K}_t^{n_p-1} \mathbf{B}_t & \mathbf{C}\mathbf{P}\mathbf{K}_t^{n_p-2} \mathbf{B}_t & \cdots & \mathbf{C}\mathbf{P}\mathbf{K}_t^{n_p-n_c} \mathbf{B}_t \end{bmatrix} \in \mathbb{R}^{q \cdot n_p \times p \cdot n_c} \quad (4.21c)$$

ergibt sich

$$\bar{\mathbf{y}}_{k+1} = \tilde{\mathbf{F}}\Psi_k + \tilde{\mathbf{H}}\bar{\mathbf{u}}_k. \quad (4.22)$$

Die quadratische Kostenfunktion

$$J(\bar{\mathbf{u}}_k) = (\bar{\mathbf{y}}_{k+1} - \bar{\mathbf{w}}_{k+1})^\top \bar{\mathbf{Q}} (\bar{\mathbf{y}}_{k+1} - \bar{\mathbf{w}}_{k+1}) + \bar{\mathbf{u}}_k^\top \bar{\mathbf{R}} \bar{\mathbf{u}}_k \quad (4.23)$$

lässt sich mit dem Fehlerterm

$$\mathbf{e}_k := \tilde{\mathbf{F}}\Psi_k - \bar{\mathbf{w}}_{k+1} \Rightarrow \bar{\mathbf{y}}_{k+1} - \bar{\mathbf{w}}_{k+1} = \mathbf{e}_k + \tilde{\mathbf{H}}\bar{\mathbf{u}}_k \quad (4.24)$$

zu

$$\begin{aligned} J(\bar{\mathbf{u}}_k) &= (\mathbf{e}_k + \tilde{\mathbf{H}}\bar{\mathbf{u}}_k)^\top \bar{\mathbf{Q}} (\mathbf{e}_k + \tilde{\mathbf{H}}\bar{\mathbf{u}}_k) + \bar{\mathbf{u}}_k^\top \bar{\mathbf{R}} \bar{\mathbf{u}}_k \\ &= \bar{\mathbf{u}}_k^\top (\tilde{\mathbf{H}}^\top \bar{\mathbf{Q}} \tilde{\mathbf{H}} + \bar{\mathbf{Q}}) \bar{\mathbf{u}}_k + 2\bar{\mathbf{u}}_k^\top \tilde{\mathbf{H}}^\top \bar{\mathbf{Q}} \mathbf{e}_k + \mathbf{e}_k^\top \bar{\mathbf{Q}} \mathbf{e}_k \end{aligned} \quad (4.25)$$

umformen, wobei die Gewichtungsmatrizen  $\bar{\mathbf{Q}} \in \mathbb{R}^{q \times q}$  und  $\bar{\mathbf{R}} \in \mathbb{R}^{p \times p}$  zu

$$\bar{\mathbf{Q}} = \begin{bmatrix} \bar{\mathbf{Q}} & & \\ & \ddots & \\ & & \bar{\mathbf{Q}} \end{bmatrix} \in \mathbb{R}^{q \cdot n_p \times q \cdot n_p}, \quad \bar{\mathbf{R}} = \begin{bmatrix} \bar{\mathbf{R}} & & \\ & \ddots & \\ & & \bar{\mathbf{R}} \end{bmatrix} \in \mathbb{R}^{p \cdot n_c \times p \cdot n_c} \quad (4.26)$$

erweitert werden.

Die Minimierung der Kostenfunktion

$$\frac{\partial J(\bar{\mathbf{u}}_k)}{\partial \bar{\mathbf{u}}_k} = 2 \left( \tilde{\mathbf{H}}^\top \bar{\mathbf{Q}} \tilde{\mathbf{H}} + \bar{\mathbf{R}} \right) \bar{\mathbf{u}}_k + 2 \tilde{\mathbf{H}}^\top \bar{\mathbf{Q}} \mathbf{e}_k = \mathbf{0}, \quad (4.27)$$

ergibt die Lösung

$$\bar{\mathbf{u}}_k = - \left( \tilde{\mathbf{H}}^\top \bar{\mathbf{Q}} \tilde{\mathbf{H}} + \bar{\mathbf{R}} \right)^{-1} \tilde{\mathbf{H}}^\top \bar{\mathbf{Q}} \mathbf{e}_k. \quad (4.28)$$

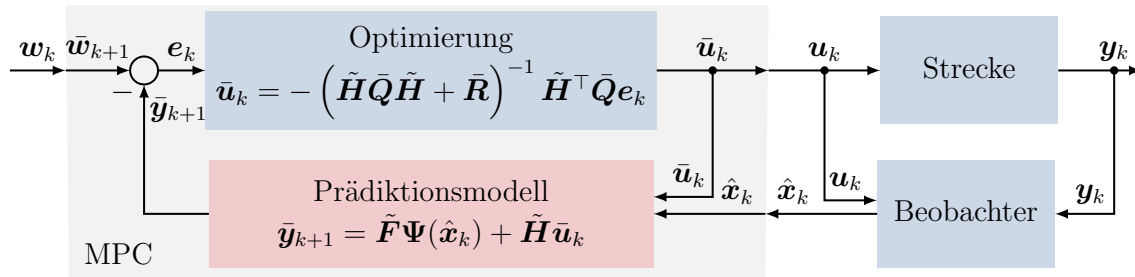
Eventuelle Stellgrößen- und Ausgangsgrößenbeschränkungen

$$\bar{\mathbf{u}}_{\min} \leq \bar{\mathbf{u}} \leq \bar{\mathbf{u}}_{\max}, \quad \bar{\mathbf{y}}_{\min} \leq \bar{\mathbf{y}} \leq \bar{\mathbf{y}}_{\max} \quad (4.29)$$

können während der numerischen Minimierung der Kostenfunktion (4.27) berücksichtigt werden. Für die Regelung wird jeweils nur der erste Wert

$$\mathbf{u}_k = [\mathbf{I}_p, \mathbf{0}, \dots, \mathbf{0}] \bar{\mathbf{u}}_k \quad (4.30)$$

als Stellgröße verwendet. In der Abbildung 4.3 wird die Struktur der EDMD-basierten MPC veranschaulicht.



**Abbildung 4.3** Struktur der EDMD-basierten modellprädiktiven Regelung.

Das zu lösende Optimierungsproblem ist konvex. Durch die Offline-Berechnung der Matrizen  $\tilde{\mathbf{F}}$  und  $\tilde{\mathbf{H}}$  ist der Rechenaufwand für die Stellgrößenfolge bei der EDMD-basierten MPC vergleichbar mit einer linearen MPC für das entsprechende klassisch linearisierte Originalsystem [47]. In der Literatur finden sich vielfältige Anwendungsfälle, beispielsweise die Regelung künstlicher pneumatischer Muskeln [147], [148], [149], gekoppelter Pendelsysteme [150], Piezo-Aktoren zur Nano-Positionierung [151], [152], [153]. Außerdem wurde die Koopman-basierte MPC zur Stabilisierung von Multicoptern [154], [155], [156] sowie im Bereich der Regelung von Fahrzeugen [157], Schiffen [158] und Hochgeschwindigkeitszügen [159] demonstriert. Darüber hinaus finden sich in der Literatur Anwendungen auf ein Viertanksystem [160], Fracking von Erdöl [161] sowie einen verfahrenstechnischen mikrobiellen Fermentationsprozess [162]. Die Autor\*innen in [163] erweitern den Ansatz auf partielle Differentialgleichungen.

EDMD-Modelle lassen sich geradlinig für den Entwurf von **modellprädiktiven Regelungen** verwenden. Aufgrund der linearen Systemstruktur von EDMD-Modellen ergibt sich ein konvexes Optimierungsproblem, dessen Lösung sich einfach implementieren lässt und eine sehr hohe Regelungsgüte liefert.

### 4.3 Passivitätsbasierte Regelung mittels datengetriebener PCHD-Modelle

Im Folgenden wird zunächst die Idee der passivitätsbasierten Regelung allgemein beschrieben. Danach wird die Anwendbarkeit der im Abschnitt 3.4 entwickelten hybriden PCHD-Modelle für einen solchen Regelungsentwurf vorgestellt und im Rahmen einer Machbarkeitsstudie beispielhaft am Schlagmechanismus des Golfroboters demonstriert.

Die passivitätsbasierte Regelung verfolgt das Ziel, die Passivitätseigenschaft einer Strecke systematisch zu nutzen, um stabile Regelkreise zu erhalten. Das Konzept wurde erstmalig in [164] eingeführt und folgend weiterentwickelt. Die Regelung mittels IDA (Interconnection and Damping Assignment) [165], [166] verfolgt das Ziel, für ein möglicherweise, aber nicht zwingend passives System

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}) + \mathbf{B}(\mathbf{x})\mathbf{u}, \quad (4.31a)$$

$$\mathbf{y} = \mathbf{c}(\mathbf{x}) \quad (4.31b)$$

eine Regelung  $\mathbf{u}(\mathbf{x})$  zu entwerfen, sodass der geschlossene Regelkreis die Dynamik eines PCHD-Systems

$$\dot{\mathbf{x}} = (\mathbf{J}_d(\mathbf{x}) - \mathbf{D}_d(\mathbf{x})) \left( \frac{\partial V_d}{\partial \mathbf{x}} \right)^\top + \mathbf{B}(\mathbf{x})\mathbf{w}_{\text{PCHD}}, \quad (4.32a)$$

$$\mathbf{y}_{\text{PCHD}} = \mathbf{B}^\top(\mathbf{x}) \left( \frac{\partial V_d}{\partial \mathbf{x}} \right)^\top \quad (4.32b)$$

mit dem kollokierten Ausgangsvektor  $\mathbf{y}_{\text{PCHD}}$  und der Führungsgröße  $\mathbf{w}_{\text{PCHD}}$  aufweist [21]. Die Matrizen  $\mathbf{J}_d(\mathbf{x}) = -\mathbf{J}_d^\top(\mathbf{x})$  und  $\mathbf{D}_d(\mathbf{x}) = \mathbf{D}_d^\top(\mathbf{x}) \succeq 0$  sowie die positiv semidefinite Funktion  $V_d : \mathbb{R}^n \rightarrow \mathbb{R}$ , die ein striktes lokales Minimum an der Stelle der gewünschten Ruhelage  $\mathbf{x}^*$  hat, charakterisieren die gewünschte Dynamik des Regelkreises.

Zur Bestimmung eines Regelgesetzes  $\mathbf{u}(\mathbf{x})$  für die Regelstrecke (4.31) mit dem Ziel, dass der Regelkreis die PCHD-Struktur (4.32) aufweist, wird die Stellgröße in zwei Komponenten aufgeteilt  $\mathbf{u} = \mathbf{u}_1 + \mathbf{u}_2$ , wobei  $\mathbf{u}_1 = \mathbf{w}_{\text{PCHD}}$  den Einfluss der Führungsgrößen und  $\mathbf{u}_2$  den Einfluss des stabilisierenden Reglers repräsentieren. Das Gleichsetzen der Gleichungen (4.31a) und (4.32a) unter der Berücksichtigung von  $\mathbf{u} = \mathbf{u}_1 + \mathbf{u}_2$  ergibt

$$\mathbf{B}(\mathbf{x})\mathbf{u}_2 = (\mathbf{J}_d(\mathbf{x}) - \mathbf{D}_d(\mathbf{x})) \left( \frac{\partial V_d}{\partial \mathbf{x}} \right)^\top - \mathbf{f}(\mathbf{x}). \quad (4.33)$$

Zur Bestimmung der Größen  $V_d(\mathbf{x})$ ,  $\mathbf{J}_d$ ,  $\mathbf{D}_d$  ist die partielle Differentialgleichung<sup>5</sup>

$$\mathbf{B}^\perp(\mathbf{x}) \left( (\mathbf{J}_d(\mathbf{x}) - \mathbf{D}_d(\mathbf{x})) \left( \frac{\partial V_d}{\partial \mathbf{x}} \right)^\top - \mathbf{f}(\mathbf{x}) \right) = \mathbf{0} \quad (4.34)$$

<sup>5</sup>Die Matrix  $\mathbf{B}^\perp(\mathbf{x}) \in \mathbb{R}^{(n-p) \times n}$  ist eine Matrix mit dem Rang  $n - p$ , die für den jeweiligen Anwendungsfall so konstruiert werden muss, dass  $\mathbf{B}^\perp(\mathbf{x})\mathbf{B}(\mathbf{x}) = \mathbf{0}$  gilt [21].



zu erfüllen. Hierfür gibt es unterschiedliche Ansätze [167]:

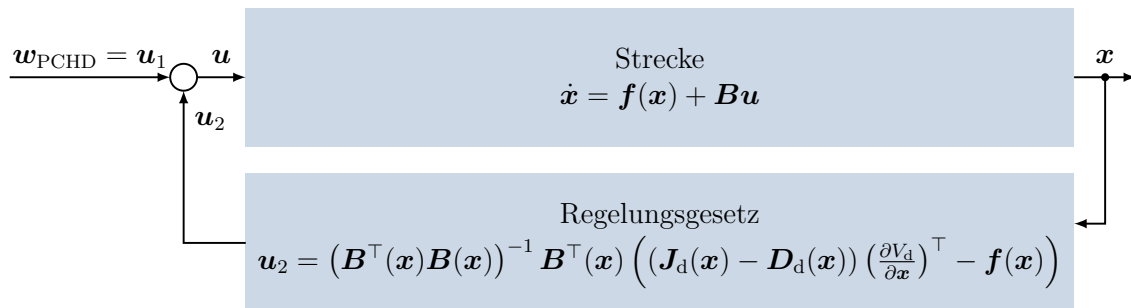
- **Algebraischer Ansatz.** Die Energiefunktion  $V_d(\mathbf{x})$  wird explizit vorgegeben. Anschließend wird Gleichung (4.34) als algebraische Gleichung in  $\mathbf{J}_d(\mathbf{x})$ ,  $\mathbf{D}_d(\mathbf{x})$  und  $\mathbf{B}^\perp(\mathbf{x})$  gelöst.
- **Nichtparametrischer Ansatz.**  $\mathbf{J}_d(\mathbf{x})$ ,  $\mathbf{D}_d(\mathbf{x})$  und  $\mathbf{B}^\perp(\mathbf{x})$  werden explizit vorgegeben und  $V_d(\mathbf{x})$  über die partielle Differentialgleichung (4.34) bestimmt.
- **Parametrischer Ansatz.** Ähnlich wie beim nichtparametrischen Ansatz wird  $V_d(\mathbf{x})$  über Gleichung (4.34) bestimmt. Allerdings erfolgt dies unter der Beschränkung, dass nur eine spezielle Klasse an Funktionen zugelassen ist. Zum Beispiel kann es bei mechanischen Systemen wünschenswert sein, die Summe der potentiellen oder kinetischen Energie des betrachteten Systems vorzugeben.

Zur Vereinfachung der Parametrierung wird in [168], [169] eine systematische Vorgehensweise durch Zuweisen einer lokal linearen Dynamik vorgeschlagen. Die Autor\*innen in [170] beschreiben hingegen, wie die Funktion  $V_d(\mathbf{x})$  zur Realisierung einer gewünschten Dynamik des geschlossenen Regelkreises mittels eines neuronalen Netzes anstelle der Lösung der partiellen Differentialgleichung approximiert werden kann.

Nach erfolgter Bestimmung der Größen  $V_d(\mathbf{x})$ ,  $\mathbf{J}_d(\mathbf{x})$ ,  $\mathbf{D}_d(\mathbf{x})$  ist das Regelungsgesetz gegeben durch [21]

$$\mathbf{u}(\mathbf{x}) = \mathbf{w}_{\text{PCHD}} + (\mathbf{B}^\top(\mathbf{x})\mathbf{B}(\mathbf{x}))^{-1} \mathbf{B}^\top(\mathbf{x}) \left( (\mathbf{J}_d(\mathbf{x}) - \mathbf{D}_d(\mathbf{x})) \left( \frac{\partial V_d}{\partial \mathbf{x}} \right)^\top - \mathbf{f}(\mathbf{x}) \right). \quad (4.35)$$

Das vorgestellte passivitätsbasierte Entwurfsverfahren liefert Regler, die stabil und robust gegenüber Parameterunsicherheiten sind. In der Abbildung 4.4 ist die grundlegende Struktur des IDA-Regelungsansatzes dargestellt.



**Abbildung 4.4** Die passivitätsbasierte Regelung mit IDA erzeugt einen passiven Regelkreis mit guter Dämpfung und einer stabilen Ruhelage.

### 4.3.1 Beispielhafte Anwendung

Im Folgenden wird demonstriert, wie sich datengetriebene PCHD-Modelle, vgl. Abschnitt 3.4, für den Entwurf passivitätsbasierter Regelungen verwenden lassen. Zur

Veranschaulichung wird der Entwurf einer IDA-Regelung für den Schlagmechanismus des Golfroboters, vgl. Abbildung 6.1, beschrieben. Als Streckenmodell wird ein datengetriebenes PCHD-Modell verwendet, das anhand simulativ erzeugter Messdaten aus dem Modell (6.11) bestimmt wurde. Mit der Energiefunktion  $V : \mathbb{R}^n \rightarrow \mathbb{R}$

$$V(\mathbf{x}) = \frac{1}{2} J x_2^2 + m g a (1 - \cos x_1) \quad (4.36)$$

und dem Gradienten  $\Psi : \mathbb{R}^n \rightarrow \mathbb{R}^n$

$$\Psi(\mathbf{x}) = \left( \frac{\partial V}{\partial \mathbf{x}} \right)^\top = \begin{bmatrix} m g a \sin x_1 \\ J x_2 \end{bmatrix} \quad (4.37)$$

ergeben sich die Matrizen

$$\mathbf{J} = \begin{bmatrix} 0 & 6,935 \\ -6,935 & 0 \end{bmatrix} =: \begin{bmatrix} 0 & j_1 \\ -j_1 & 0 \end{bmatrix}, \quad \mathbf{b} = \begin{bmatrix} 0 \\ 26,534 \end{bmatrix} =: \begin{bmatrix} 0 \\ b_2 \end{bmatrix}, \quad (4.38a)$$

$$\mathbf{D}_\succeq = \begin{bmatrix} 4,318 \cdot 10^{-5} & 0,014 \\ 0,014 & 4,610 \end{bmatrix} =: \begin{bmatrix} d_1 & d_2 \\ d_2 & d_3 \end{bmatrix}. \quad (4.38b)$$

Für die Parametrierung des IDA-Reglers wird

$$V_d(\mathbf{x}) = V(\mathbf{x}) \quad (4.39)$$

festgelegt und anschließend der algebraische Ansatz mit den parametrisierten Matrizen

$$\mathbf{J}_d = \begin{bmatrix} 0 & k_j \\ -k_j & 0 \end{bmatrix}, \quad \mathbf{D}_d = \begin{bmatrix} k_{d_1} & k_{d_2} \\ k_{d_2} & k_{d_3} \end{bmatrix}, \quad \mathbf{b}^\perp = [1, 0] \quad (4.40)$$

verfolgt. Damit ergibt sich die Schlüsselgleichung (4.34) zu

$$\mathbf{b}^\perp \left( (\mathbf{J}_d - \mathbf{J} + \mathbf{D} - \mathbf{D}_d) \left( \frac{\partial V}{\partial \mathbf{x}} \right)^\top \right) = 0 \quad (4.41)$$

bzw. zu

$$[1, 0] \begin{bmatrix} d_1 - k_{d_1} & k_j - j_1 + d_2 - k_{d_2} \\ -k_j + j_1 + d_2 - k_{d_2} & d_3 - k_{d_3} \end{bmatrix} \begin{bmatrix} m g a \sin x_1 \\ J x_2 \end{bmatrix} = 0 \quad (4.42a)$$

$$\Leftrightarrow [1, 0] \begin{bmatrix} (d_1 - k_{d_1}) m g a \sin x_1 + (k_j - j_1 + d_2 - k_{d_2}) J x_2 \\ (-k_j + j_1 + d_2 - k_{d_2}) m g a \sin x_1 + (d_3 - k_{d_3}) J x_2 \end{bmatrix} = 0. \quad (4.42b)$$

Daraus folgt

$$k_{d_1} = d_1, \quad (4.43a)$$

$$k_j - j_1 + d_2 - k_{d_2} = 0 \Leftrightarrow k_j = j_1 - d_2 + k_{d_2}. \quad (4.43b)$$

### 4.3 Passivitätsbasierte Regelung mittels datengetriebener PCHD-Modelle

Die Parameter  $k_{d_2}$  und  $k_{d_3}$  können frei gewählt werden. Damit jedoch  $\mathbf{D}_d \succeq 0$  erfüllt ist, muss

$$k_{d_3} \geq \frac{k_{d_2}^2}{d_1} \quad (4.44)$$

gelten. Beide Parameter lassen sich mittels der Gleichungen (6.16)-(6.17) numerisch optimieren zu

$$k_{d_2} = -0,1747, \quad k_{d_3} = 706,6345. \quad (4.45)$$

Als stabilisierendes Regelungsgesetz ergibt sich mit Gleichung (4.35)

$$u_2 = \frac{1}{b_2} (2(d_2 - k_{d_2}) m g a \sin x_1 + (d_3 - k_{d_3}) J x_2). \quad (4.46)$$

Für den Systemausgang gilt nach Gleichung (4.32)

$$y_{\text{PCHD}} = \mathbf{b}^\top \left( \frac{\partial V}{\partial \mathbf{x}} \right)^\top = b_2 J x_2 \quad (4.47)$$

und damit für die neue Führungsgröße

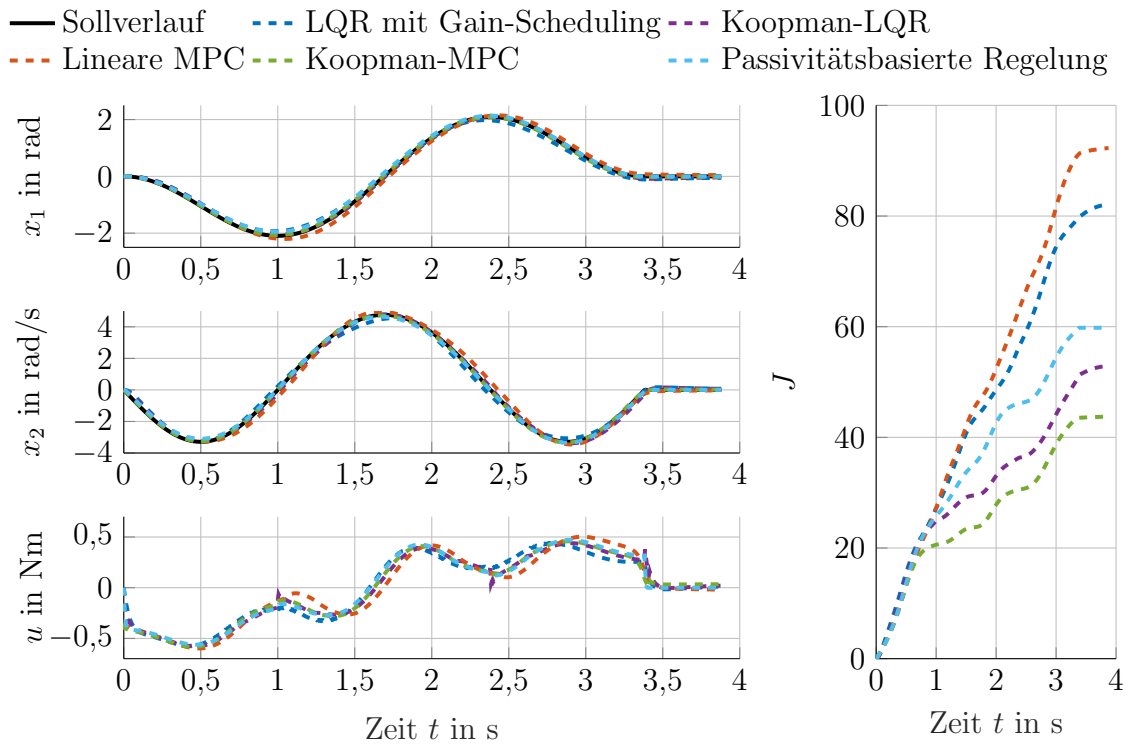
$$w_{\text{PCHD}} = b_2 J w_2, \quad (4.48)$$

sodass sich schließlich das resultierende Regelungsgesetz

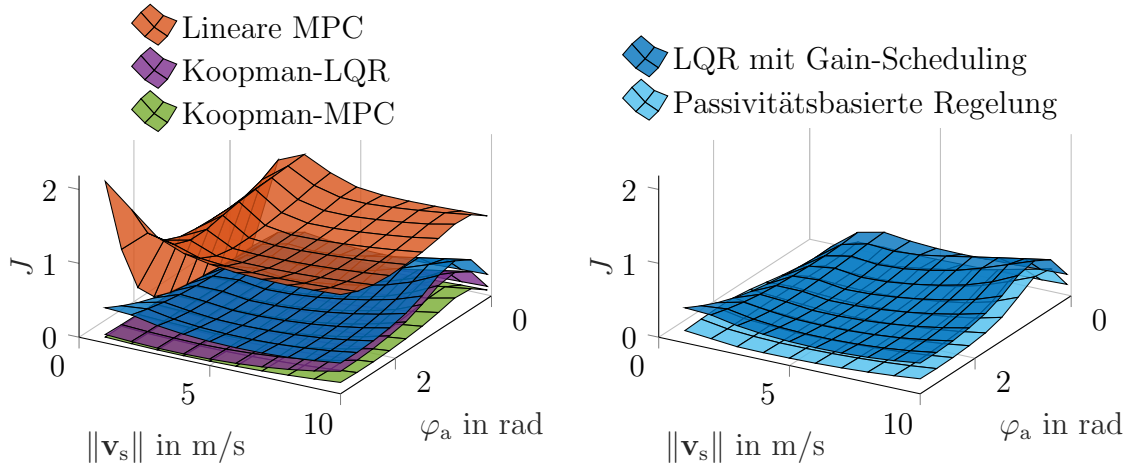
$$u = u_1 + u_2 = b_2 J w_2 + \frac{1}{b_2} (2(d_2 - k_{d_2}) m g a \sin x_1 + (d_3 - k_{d_3}) J x_2) \quad (4.49)$$

ergibt. Die Abbildung 4.5(a) zeigt die simulierte resultierende Regelgüte der passivitätsbasierten Regelungsansatzes, die mittels Gleichung (6.16)-(6.17) berechnet wurde, im Vergleich zur bisher verwendeten LQ-Regelung mit Gain-Scheduling. Es ist zu erkennen, dass der passivitätsbasierte Ansatz mit dem hybriden PCHD-Modell stark überlegen ist. Darüber hinaus visualisiert die Abbildung 4.5(c) die Regelgüte in Abhängigkeit der frei wählbaren Parameter der Solltrajektorie, vgl. Tabelle 6.1. Der passivitätsbasierte Ansatz mit dem hybriden PCHD-Modell erweist sich als überlegen gegenüber der bisher verwendeten LQ-Regelung mit Gain-Scheduling für alle wählbaren Werte, wobei der Aufwand bei der Modellbildung stark reduziert ist. Die Methode, ein PCHD-Modell datengetrieben zu bestimmen und systematisch für den Entwurf einer passivitätsbasierten Regelung zu verwenden, wurde erstmals in den Beiträgen [16] und [123] entwickelt und validiert.

Datengetriebene PCHD-Modelle lassen sich systematisch für den **passivitätsbasierten Regelungsentwurf** nutzen. Basierend auf dem überaus nützlichen Konzept der Passivität lassen sich **robuste global stabile Regelkreise mit gewünschten dynamischen Eigenschaften** entwerfen. Die resultierende Regelungsgüte war für das betrachtete Beispiel zwar geringfügig schlechter als die der EDMD-basierten Ansätze, das Modell bietet dafür jedoch eine **sehr hohe physikalische Interpretierbarkeit**.



(a) Resultierende simulierte Regelgüte der unterschiedlichen Regelungsansätze.



(b) Analyse der Koopman-basierten Regelungen.

(c) Analyse der passivitätsbasierten Regelung.

**Abbildung 4.5** Simulationsbasierte Analyse der unterschiedlichen in diesem Kapitel beschriebenen Regelungsansätze im Vergleich zur bisher verwendeten Riccati-Regelung mit Gain-Scheduling. In den Abbildungen 4.5(b) und 4.5(c) ist die mittlere resultierende Regelgüte  $J$  jeweils in Abhängigkeit der frei wählbaren Parameter  $\varphi_a$  (Ausholwinkel) und  $\|v_s\|$  (Schlaggeschwindigkeit) dargestellt.

## 4.4 Entwurfsmethoden im Kontext der Gauß-Prozess-Regression

In diesem Abschnitt werden regelungstechnische Entwurfsmethoden im Kontext der Gauß-Prozess-Regression (siehe auch Abschnitt 3.5) vorgestellt. Ebenso wie bei der hybriden Modellierung ist hierbei eine wesentliche Charakteristik die Einbindung der Wahrscheinlichkeitstheorie. Die Vorteile eines probabilistischen Ansatzes für den Steuerungs- und Regelungsentwurf sind vielfältig. Insbesondere vor dem Hintergrund der Inbetriebnahme lassen sich zwei wesentliche Vorteile nennen. Zum einen findet die Inbetriebnahme in Anlehnung an das V-Modell nach [171] iterativ statt und hat als Ziel, ein sicheres und funktionsfähiges System zu gewährleisten. Diese Iterationen sind mit einem Lernvorgang beim Reinforcement Learning vergleichbar, wobei das sogenannte Exploitation-Exploration-Dilemma [9] adressiert werden muss. In jeder Iteration muss dabei ein Kompromiss gefunden werden, bei welchem bspw. der Regler erprobt wird, welcher sowohl am erfolgversprechendsten (Exploitation) ist, als auch neue Erkenntnisse über das Systemverhalten (Exploration) liefert, siehe auch RL in Abschnitt 2.2. In diesem Sinne ist die Einbringung der Wahrscheinlichkeitstheorie besonders sinnvoll, da über sie eine zusätzliche Informationsquelle über die Unsicherheit zugänglich wird, welche zur Adressierung des Exploitation-Exploration-Dilemma dient. In [126] werden bspw. ein deterministischer und probabilistischer Lernvorgang gegenübergestellt und gezeigt, dass der wahrscheinlichkeitsbasierte Ansatz essentiell notwendig für die Erfüllung der Aufgabe ist. Ein weiterer Vorteil bei der Inbetriebnahme ist, dass qualitativ hochwertigere Aussagen über das teilweise unbekannte Systemverhalten getroffen werden können. Diese betreffen zum einen unerwünschte Systemüberschreitungen, welche durch technische Beschränkungen vorgegeben werden und zum anderen sicherheitskritische Instabilitäten, welche durch einen falsch ausgelegten Regler hervorgerufen werden können. Vor diesem Hintergrund liefert die Wahrscheinlichkeitstheorie ein qualitatives Maß, um die Unsicherheit über das reale System bzw. den Prüfstand geeignet zu beschreiben. Auf Basis dieser Informationsquelle lässt sich das Systemverhalten besser einschätzen und somit geeignete Sicherheitsmaßnahmen vornehmen.

### 4.4.1 Hybride Optimalsteuerung dynamischer Systeme

Auf der Grundlage des hybriden Modells  $\tilde{\mathbf{f}}(\mathbf{m}_k, \mathbf{S}_k, \mathbf{u}_k) := [\mathbf{m}_{k+1}, \mathbf{S}_{k+1}]$  aus Gleichung (3.130) wird in diesem Abschnitt ein hybrides Optimalsteuerungsverfahren entwickelt, welches für die iterative Inbetriebnahme eingesetzt wird. Eine typische Zielgröße bzw. Gütefunktion, die in der Regelungstechnik und im Zusammenhang mit einer Optimalsteuerung oft verwendet wird, ist wie folgt definiert

$$J(\boldsymbol{\theta}) = \Delta t \sum_{k=0}^H w_t(k) (\mathbf{x}_k - \mathbf{x}_G)^T \mathbf{W}_x (\mathbf{x}_k - \mathbf{x}_G) + \mathbf{u}_k^T \mathbf{W}_u \mathbf{u}_k. \quad (4.50)$$

Diese Zielfunktion enthält sowohl einen Term für die Distanz zum jeweiligen Zielzustand  $(\mathbf{x}_k - \mathbf{x}_G)$  als auch mit  $\mathbf{u}_k^T \mathbf{W}_u \mathbf{u}_k$  einen Term für den verbrauchten Energieaufwand.  $\mathbf{W}_x \succ 0$  und  $\mathbf{W}_u \succeq 0$  sind Gewichtungsmatrizen.  $w_t(k) \in [0, 1]$  steigt monoton an und gibt somit ein höheres Gewicht für Abweichungen, die in einem

späteren Verlauf der Trajektorie auftreten. Der Optimierungsvektor  $\boldsymbol{\theta}$  wird im weiteren Verlauf dieses Abschnitts definiert. Da der Zustand des hybriden Modells als normalverteilt angenommen wird, d. h.  $\mathbf{x}_k \sim \mathcal{N}(\mathbf{m}_k, \mathbf{S}_k)$  gilt, ist der Erwartungswert als tatsächliche Zielgröße zu verwenden. Die analytische Lösung der erwartbaren Gütefunktion lautet

$$\begin{aligned} \mathbb{E}_{\mathbf{x}_k}[J(\boldsymbol{\theta})] = & \Delta t \sum_{k=0}^H w_t(k) \left( \left( \sum_{i=1}^{n_x} \mathbf{S}_k^{(i,i)} \mathbf{W}_x^{(i,i)} \right) + (\mathbf{m}_k - \mathbf{x}_G)^T \mathbf{W}_x (\mathbf{m}_k - \mathbf{x}_G) \right) \\ & + \mathbf{u}_k^T \mathbf{W}_u \mathbf{u}_k. \end{aligned} \quad (4.51)$$

Neben der Gütefunktion gilt es, die technischen Rahmenbedingungen im hybriden Optimalsteuerungsproblem zu adressieren. Steuerungsbeschränkungen aufgrund von limitierter Aktorik können auf herkömmliche Weise durch  $\mathbf{u}_{min} \leq \mathbf{u}_k \leq \mathbf{u}_{max}$  berücksichtigt werden. Zustandsbeschränkungen können durch den Erwartungswert eingebunden werden, z. B. über  $\mathbf{x}_{min} \leq \mathbf{m}_k \leq \mathbf{x}_{max}$ , was jedoch aufgrund der Vernachlässigung der Varianz nicht besonders zuverlässig wäre. Daher wird ein wahrscheinlichkeitsbasierter Ansatz verfolgt, der in Hinblick auf die Inbetriebnahme geeigneter ist. Die Wahrscheinlichkeit, die Zustandsbeschränkungen unter der konservativen Annahme, dass die Dimensionen unabhängig sind, zu erfüllen, wird durch

$$\begin{aligned} P(\mathbf{x}_{min} \leq \mathbf{x}_k \leq \mathbf{x}_{max}) & \approx \prod_{i=1}^{n_x} P(x_{min}^{(i)} \leq x_k^{(i)} \leq x_{max}^{(i)}) \\ & = \prod_{i=1}^{n_x} \Phi\left(\frac{x_{max}^{(i)} - m_k^{(i)}}{\sqrt{S_k^{(i,i)}}}\right) - \Phi\left(\frac{x_{min}^{(i)} - m_k^{(i)}}{\sqrt{S_k^{(i,i)}}}\right) \geq P_x \end{aligned} \quad (4.52)$$

ausgedrückt, wobei  $\Phi(\cdot)$  die Verteilungsfunktion der Standardnormalverteilung ist und  $P_x \in (0, 1]$  ein vordefinierter Wahrscheinlichkeitsgrenzwert ist. Die Zustandsbeschränkungen sind in der Regel mit sicherheitskritischen Aspekten verbunden. Aus diesem Grund sollte der Wert von  $P_x$  relativ groß gewählt werden, damit die Zustandsbeschränkungen konservativ eingehalten werden.

An dieser Stelle wird das vollständige hybride Optimalsteuerungsproblem, mit welchem die Inbetriebnahme durchgeführt wird, definiert

$$\begin{aligned} \min_{\boldsymbol{\theta}} \mathbb{E}_{\mathbf{x}_k}[J(\boldsymbol{\theta})] \quad \text{u.B.v.} \quad & \mathbf{h}(\boldsymbol{\theta}) = \mathbf{0}, \quad \mathbf{g}(\boldsymbol{\theta}) \leq \mathbf{0}, \\ \mathbf{h}(\boldsymbol{\theta}) = & \begin{cases} [\mathbf{m}_{k+1}, \mathbf{S}_{k+1}] = \tilde{\mathbf{f}}(\mathbf{m}_k, \mathbf{S}_k, \mathbf{u}_k), & k = 0, \dots, H-1, \\ \mathbf{m}_0 - \mathbf{m}_I, & \mathbf{S}_0 - \mathbf{S}_I, \\ \mathbf{m}_H - \mathbf{x}_G, \end{cases} \\ \mathbf{g}(\boldsymbol{\theta}) = & \begin{cases} P_x - P(\mathbf{x}_{min} \leq \mathbf{x}_k \leq \mathbf{x}_{max}), \\ \mathbf{u}_k - \mathbf{u}_{max}, & \mathbf{u}_{min} - \mathbf{u}_k, \quad k = 0, \dots, H, \end{cases} \end{aligned} \quad (4.53)$$

mit einer initialen Zustandsverteilung  $\mathbf{x}_I \sim \mathcal{N}(\mathbf{m}_I, \mathbf{S}_I)$  und  $H$  der Gesamtanzahl der betrachteten diskreten Punkte. Ähnlich dem Ansatz des Multiple Shooting [172] für deterministische dynamische Modelle erzwingen die Gleichheitsbeschränkungen  $\mathbf{h}(\boldsymbol{\theta})$ , dass die probabilistischen Dynamikgleichungen des hybriden Modells (3.130)

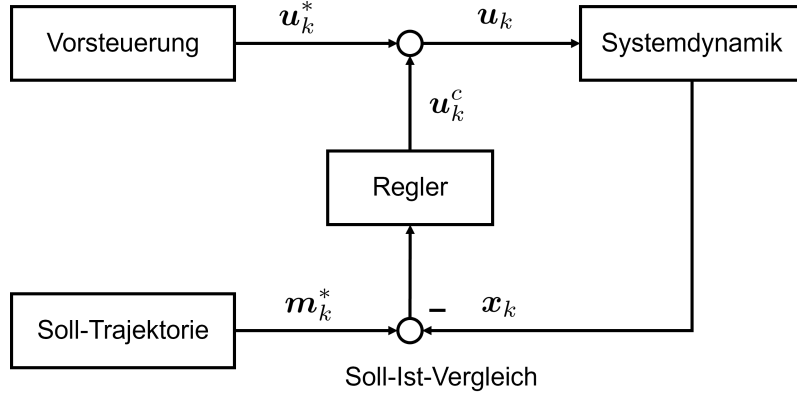


Abbildung 4.6 Zwei-Freiheitsgrade Regelungsstruktur

für alle diskreten Zeitschritte erfüllt sind. Dies erfordert, dass die propagierten Momente  $\mathbf{f}_{hyb}(\mathbf{m}_k, \mathbf{S}_k)$  bei gegebener Steuerung  $\mathbf{u}_k$  mit den nächsten Momenten  $(\mathbf{m}_{k+1}, \mathbf{S}_{k+1})$  übereinstimmen. Der Optimierungsvektor setzt sich damit aus den folgenden Größen zusammen:

$$\boldsymbol{\theta} = [\mathbf{m}_0^T, \tilde{\mathbf{L}}_0^T, \mathbf{u}_0^T, \dots, \mathbf{m}_H^T, \tilde{\mathbf{L}}_H^T, \mathbf{u}_H^T]^T \in \mathbb{R}^{\left(\frac{1}{2}n_x^2 + \frac{3}{2}n_x + n_u\right)(H+1)}, \quad (4.54)$$

wobei die Cholesky-Zerlegung mit

$$\mathbf{S}_k = \mathbf{L}_k \mathbf{L}_k^T, \quad \tilde{\mathbf{L}}_k = \text{nonzeros}(\mathbf{L}_k), \quad (4.55)$$

verwendet wird. Die Funktion  $\text{nonzeros}(\cdot)$  bildet dabei die Elemente der unteren Dreiecksmatrix von  $\mathbf{L}_k$  auf einen Spaltenvektor  $\mathbf{L}_{\Delta,k} \in \mathbb{R}^{\frac{1}{2}(n_x^2 + n_x)}$  ab. Die Formulierung über die Cholesky-Zerlegung erzwingt implizit eine weitere Nebenbedingung, bei der Symmetrie und positive Definitheit der Matrizen der Zustandsvarianzen erzwungen werden. Um das hybride Optimalsteuerungsproblem zu lösen, kann bspw. das SQP-Verfahren [173] eingesetzt werden. Die Lösung liefert dann eine probabilistische (Soll-)Trajektorie  $\{\mathbf{m}_k^*, \mathbf{S}_k^*\}$  und den zugehörigen Stellgrößenverlauf  $\{\mathbf{u}_k^*\}$  mit  $k = 0, \dots, H$ . Zur Kompensation von etwaigen Störungen wird zusätzlich eine Regelung benötigt. Hierzu wird die Zwei-Freiheitsgrade Regelungsstruktur [24] verwendet, welche sich aus einem Steuerungs- und Regelungsanteil zusammensetzt (siehe Abbildung 4.6). Die Regelung erfolgt auf die Trajektorie, welche am wahrscheinlichsten erscheint, also den Verlauf des Zustandserwartungsvektors  $\{\mathbf{m}_k^*\}$  mit  $k = 0, \dots, H$ . Der Steuerungsanteil ist  $\mathbf{u}_k^*$  und der Regelungsanteil  $\mathbf{u}_k^c$  wird durch einen zeitvarianten Riccati-Regler [174] realisiert. Die gesamte Stellgröße lautet somit  $\mathbf{u}_k = \mathbf{u}_k^* + \mathbf{u}_k^c$  und beinhaltet das Regelgesetz

$$\mathbf{u}_k^c(\mathbf{x}_k) = \mathbf{K}_k(\mathbf{m}_k^* - \mathbf{x}_k), \quad (4.56)$$

wobei sich die Verstärkungsmatrix  $\mathbf{K}_k$  für jeden Zeitschritt nach

$$\mathbf{K}_k = (\mathbf{W}_u + \mathbf{B}_k^T \mathbf{P}_k \mathbf{B}_k)^{-1} \mathbf{B}_k^T \mathbf{P}_k \mathbf{A}_k, \quad (4.57)$$

ergibt. Hierin stellt  $\mathbf{P}_k \in \mathbb{R}^{n_x \times n_x}$  die Riccati-Matrix dar, welche mittels zeitlicher

Rückwärtsintegration der riccatischen Differentialgleichung bestimmt wird. In einer bereits diskretisierten Form lautet der Zusammenhang

$$\mathbf{P}_k = \mathbf{W}_x + \mathbf{A}_k^T \mathbf{P}_{k+1} \mathbf{A}_k - \mathbf{A}_k^T \mathbf{P}_{k+1} (\mathbf{B}_k (\mathbf{W}_u + \mathbf{B}_k^T \mathbf{P}_{k+1} \mathbf{B}_k)^{-1} \mathbf{B}_k^T \mathbf{P}_{k+1}) \mathbf{A}_k, \quad (4.58)$$

$k = H - 1, \dots, 0$ , wobei sich die Anfangs- bzw. Endbedingung  $\mathbf{P}_H$  aus der Lösung der statischen Riccatigleichung ergibt. Des Weiteren sind  $\mathbf{A}_k \in \mathbb{R}^{n_x \times n_x}$  und  $\mathbf{B}_k \in \mathbb{R}^{n_x \times n_u}$  die Dynamik- und Eingangsmatrix, welche sich aus einer Linearisierung des hybriden Modells ergeben. Aufgrund der gewählten additiven Zusammensetzung des hybriden Modells (3.130) lassen sich die entsprechenden Ableitungen des physikalischen und datengetriebenen Modellteils separat voneinander bilden über

$$\mathbf{A}_k = \left. \frac{\partial \hat{\mathbf{f}}}{\partial \mathbf{x}_k} \right|_{\substack{\mathbf{x}_k = \mathbf{m}_k^* \\ \mathbf{u}_k = \mathbf{u}_k^*}} + \left. \frac{\partial \Delta \mu}{\partial \mathbf{x}_k} \right|_{\substack{\mathbf{x}_k = \mathbf{m}_k^* \\ \mathbf{u}_k = \mathbf{u}_k^*}}, \quad \mathbf{B}_k = \left. \frac{\partial \hat{\mathbf{f}}}{\partial \mathbf{u}_k} \right|_{\substack{\mathbf{x}_k = \mathbf{m}_k^* \\ \mathbf{u}_k = \mathbf{u}_k^*}} + \left. \frac{\partial \Delta \mu}{\partial \mathbf{u}_k} \right|_{\substack{\mathbf{x}_k = \mathbf{m}_k^* \\ \mathbf{u}_k = \mathbf{u}_k^*}}. \quad (4.59)$$

Da die Stellgröße nun über das Regelgesetz direkt von der Zustandsgröße abhängt, muss diese ebenso als Zufallsvariable aufgefasst werden. Da es sich bei dem Regelgesetz um eine lineare Beziehung handelt, kann die Wahrscheinlichkeitsverteilung der Stellgröße direkt über den Zusammenhang

$$\begin{aligned} \mathbf{u}_k &\sim \mathcal{N}(\mathbf{m}_{u_k}, \mathbf{S}_{u_k}), \\ \mathbf{m}_{u_k} &= \mathbf{u}_k^* + \mathbf{K}_k (\mathbf{m}_k^* - \mathbf{m}_{x_k}), \\ \mathbf{S}_{u_k} &= \mathbf{K}_k \mathbf{S}_{x_k} \mathbf{K}_k^T, \end{aligned} \quad (4.60)$$

angegeben werden. Die probabilistische Stellgröße wirkt sich über die Systemdynamik auf den probabilistischen Zustandsgrößenverlauf aus, welcher mit Hilfe der besprochenen UT (siehe Unterabschnitt 3.5.2) berechnet werden kann. Vor diesem Hintergrund ist es sinnvoll den offenen und geschlossenen Regelkreis formal zu kennzeichnen. Die Prädiktion des offenen Regelkreises ergibt sich aus der Lösung des hybriden Optimalsteuerungsproblems und lautet  $\{\mathbf{m}_k^*, \mathbf{S}_k^*\}$  für  $\{\mathbf{u}_k^*\}$  mit  $k = 0, \dots, H$ . Die Prädiktion des geschlossenen Regelkreises lautet  $\{\mathbf{m}_{g,k}, \mathbf{S}_{g,k}\}$  für  $\mathbf{u}_k = \mathbf{u}_k^* + \mathbf{K}_k (\mathbf{m}_k^* - \mathbf{x}_k)$ .

Der Algorithmus der hybriden Optimalsteuerung, welcher zur Inbetriebnahme bei partieller Systemkenntnis eingesetzt wird, ist im Algorithmus 1 zusammengefasst. Aus einer übergeordneten Perspektive verwendet die Methodik einen konventionellen Trial-and-Error-Ansatz. Der Prozess beginnt mit der Lösung des Optimalsteuerungsproblems, das ausschließlich auf dem etablierten physikalischen Modell basiert. Anschließend wird die aktuelle optimale Steuersequenz auf das reale System angewendet und dabei Zustandsdaten aufgezeichnet und gesammelt (Zeile 1). Nachfolgend wird eine Zwei-Schritt-Schleife ausgeführt (Zeilen 2-10 und 4-6). Die äußere Schleife beginnt mit dem Erlernen der Modellfehler im Zusammenhang mit dem physikalischen Modellteil unter Verwendung der Gauß-Prozess-Regression und aller verfügbaren Daten (Zeile 3). Danach wird die innere Schleife ausgelöst und das hybride Optimalsteuerungsproblem basierend auf dem hybriden Modell bis zur Konvergenz gelöst (Zeilen 4-6). Dies beinhaltet die wiederholte Berechnung der Ableitungen, wofür die automatische Differenzierung genutzt werden kann, sowie die Aktualisierung der



---

**Algorithmus 1** Hybride Optimalsteuerung
 

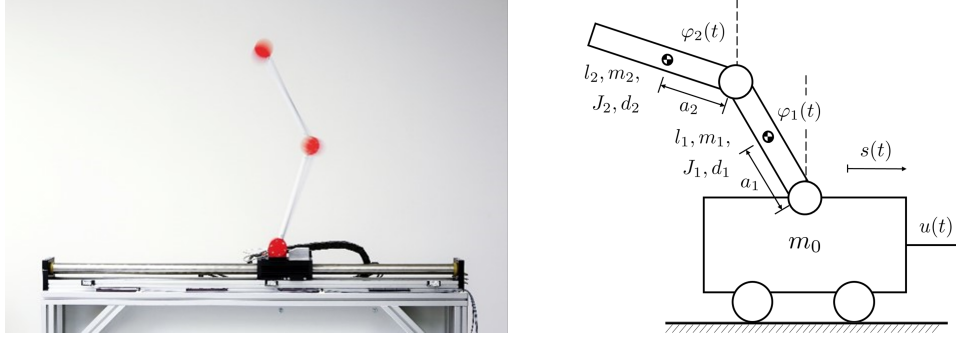
---

- 1: **Initialisierung:** Berechne Optimalsteuerung  $\{\mathbf{u}_k^*\}_{k=0}^H$  auf Basis des physikalischen Modells  $\hat{\mathbf{f}}$  und teste sie am realen System, wobei initiale Messdaten gesammelt werden.
  - 2: **Wiederhole** bis ein Konvergenzkriterium oder ein Iterationsbudget erreicht ist:
  - 3:   Lerne Modellfehler über GPs (3.116) auf Basis der vorhandenen Daten (3.114).
  - 4:   **Wiederhole** die nachfolgenden Schritte im Rahmen einer SQP-Optimierung
  - 5:     Berechne den Gradienten  $\nabla \mathbb{E}[J(\boldsymbol{\theta})]$  und die Jacobimatrizen der Nebenbedingungen  $\nabla \mathbf{h}(\boldsymbol{\theta}), \nabla \mathbf{g}(\boldsymbol{\theta})$  ▷ Automatische Differenzierung
  - 6:     Aktualisiere den Optimierungsvektor  $\boldsymbol{\theta}$  und die zugehörigen Lagrange-Multiplikatoren  $(\boldsymbol{\lambda}, \boldsymbol{\mu})$ .
  - 7:   Erhalte Optimalsteuerung und Prädiktion (offener Regelkreis)  $\{\mathbf{m}_k^*, \mathbf{S}_k^*, \mathbf{u}_k^*\}_{k=0}^H$ .
  - 8:   Berechne zeitvarianten Riccati-Regler  $\{\mathbf{K}_k\}_{k=0}^H$  nach Gleichung (4.57), (4.58) und Prädiktion für den geschlossenen Regelkreis  $\{\mathbf{m}_{g,k}, \mathbf{S}_{g,k}\}_{k=0}^H$ .
  - 9:   **Falls**  $S_{g,k}^{(i,i)} \leq S_{max}^{(i,i)}, k = 0, \dots, H, i = 1, \dots, n_x$  dann ▷ Überprüfe Stabilität  
     setze  $\mathbf{u}_k \leftarrow \mathbf{u}_k^* + \mathbf{K}_k(\mathbf{m}_k^* - \mathbf{x}_k)$ ,  
   **sonst**  
     setze  $\mathbf{u}_k \leftarrow \mathbf{u}_k^*$ .
  - 10:   Erprobe  $\mathbf{u}_k$  am realen System und nehme weitere Messdaten auf, die den bestehenden Daten hinzugefügt werden.
- 

Optimierungsvariablen und vorhandener Lagrange-Multiplikatoren. In den Zeilen 7 und 8 wird der offene und geschlossene Regelkreis mittel UT berechnet, wobei der zeitvariante Riccati-Regler durch die angesprochene Linearisierung des hybriden Modells aufgestellt wird. Anschließend erfolgt eine Überprüfung der Stabilität, wobei die Zustandsvarianz  $\mathbf{S}_{g,k}$  mit einem vordefinierten Grenzwert  $\mathbf{S}_{max}$  verglichen wird. Ist die Stabilität des geschlossenen Regelkreises dadurch gewährleistet, wird der Regler am realen System eingesetzt. Ist sie nicht gegeben, wird lediglich die Optimalsteuerung verwendet. Bei der Erprobung (Zeile 10) werden weitere Messdaten gesammelt und in jeder Iteration den bestehenden Messdaten hinzugefügt. Der sich abwechselnde Lern-, Verbesserungs- und Interaktionszyklus wird wiederholt, bis die Inbetriebnahme erfolgreich abgeschlossen ist.

**Anwendung auf ein reales Doppelpendel auf einem Wagen** Weiterführend wird nun Algorithmus 1 für ein Doppelpendel auf einem Wagen beispielhaft eingesetzt. Abbildung 4.7 zeigt hierzu den Prüfstand und die zugehörige Prinzipskizze auf deren Basis eine physikalische Modellbildung durchgeführt wird. Die vollständigen Bewegungsgleichungen in einer partiell zustandslinearisierten Form [116], welche durch den Lagrange-Formalismus aufgestellt werden können, lauten

$$\begin{aligned} \mathcal{M}_1(\varphi)\ddot{\varphi} &= \mathcal{F}(\dot{\varphi}) - \mathcal{C}(\varphi, \dot{\varphi}) - \mathcal{G}(\varphi) - \mathcal{M}_2(\varphi)u, \\ \ddot{s} &= u, \end{aligned} \tag{4.61}$$



**Abbildung 4.7** Doppelpendel auf einem Wagen: Prüfstand (links) und zugehörige Prinzipskizze (rechts) mit relevanten Zustandsgrößen und mechanischen Parametern.

mit Winkelvektor  $\varphi = [\varphi_1, \varphi_2]^T$  und den klassischen mechanischen Systemmatrizen

$$\begin{aligned}\mathcal{M}_1 &= \begin{bmatrix} J_1 + a_1^2 m_1 + l_1^2 m_2 & a_2 l_1 m_2 \cos(\varphi_1 - \varphi_2) \\ a_2 l_1 m_2 \cos(\varphi_1 - \varphi_2) & J_2 + a_2^2 m_2 \end{bmatrix}, \\ \mathcal{M}_2 &= \begin{bmatrix} (l_1 m_2 + a_1 m_1) \cos(\varphi_1) \\ a_2 m_2 \cos(\varphi_2) \end{bmatrix}, \\ \mathcal{F} &= \begin{bmatrix} d_2(\dot{\varphi}_2 - \dot{\varphi}_1) - d_1 \dot{\varphi}_1 \\ d_2(\dot{\varphi}_1 - \dot{\varphi}_2) \end{bmatrix}, \\ \mathcal{C} &= \begin{bmatrix} a_2 l_1 m_2 \sin(\varphi_1 - \varphi_2) \dot{\varphi}_2^2 \\ a_2 l_1 m_2 \sin(\varphi_1 - \varphi_2) \dot{\varphi}_1^2 \end{bmatrix}, \\ \mathcal{G} &= \begin{bmatrix} -g(a_1 m_1 + l_1 m_2) \sin(\varphi_1) \\ -g a_2 m_2 \sin(\varphi_2) \end{bmatrix}.\end{aligned}\quad (4.62)$$

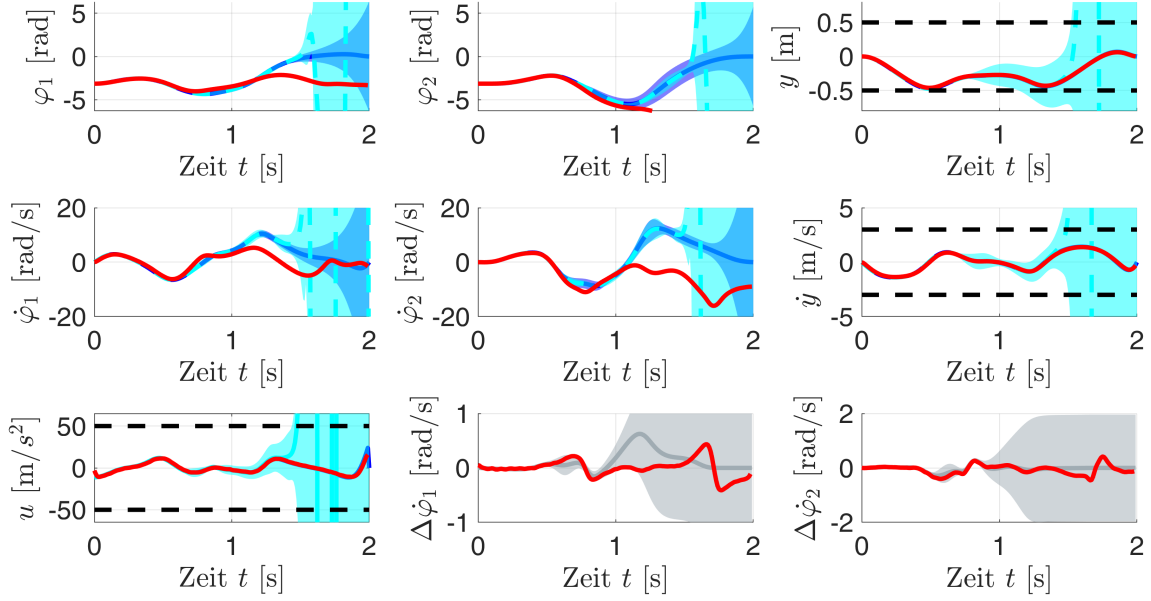
Für das betrachtete Lernszenario wird das nichtlineare Zustandsraummodell

$$\hat{\mathbf{f}}(\mathbf{x}_k, u_k) = \mathbf{x}_k + \Delta t \begin{bmatrix} \dot{\varphi}_k \\ \dot{s}_k \\ \mathcal{M}_1(\varphi_k)^{-1}(\mathcal{F}(\dot{\varphi}_k) - \mathcal{G}(\varphi_k) - \mathcal{M}_2(\varphi_k)u_k) \\ u_k \end{bmatrix}, \quad (4.63)$$

mit  $\mathbf{x} = [\varphi_1, \varphi_2, s, \dot{\varphi}_1, \dot{\varphi}_2, \dot{s}]^T$  als Zustandsvektor, als bekannt vorausgesetzt. Beim Vergleich mit Gleichung (4.62) fällt auf, dass die relevanten Zentrifugalkräfte fehlen, womit diese als unbekannte Modellfehler über

$$\Delta \mathbf{x}_{k+1}(\varphi_k, \dot{\varphi}_k) = \begin{bmatrix} \mathbf{0} \\ -\Delta t \mathcal{M}_1(\varphi_k)^{-1} \mathcal{C}(\varphi_k, \dot{\varphi}_k) \dot{\varphi}_k \\ \mathbf{0} \end{bmatrix}, \quad (4.64)$$

definiert werden. Die Aufgabe besteht darin, die Pendelarme aus der unteren in die obere Position durch eine geeignete Bewegung des Wagens zu bringen. Ohne die Kenntnis der Zentrifugalkräfte ist die Inbetriebnahme und die Lösung der Aufgabe nicht möglich, wodurch der Einsatz der hybriden Optimalsteuerung motiviert wird. Da in Gleichung (4.64) einige Elemente 0 entsprechen, ist eine Korrektur durch den datengetriebenen GP-Teil nicht notwendig. Entsprechend der Überlegungen in Unterabschnitt 3.5.2 wird eine Kopplungsmatrix  $\mathbf{B} = [\mathbf{0}^{(2 \times 3)}, \mathbf{I}^{(2 \times 2)}, \mathbf{0}^{(2 \times 1)}]^T$  eingeführt,



**Abbildung 4.8** Ergebnisse der 3. Lerniteration der hybriden Optimalsteuerung am Doppelpendel auf einem Wagen. Die Prädiktion ( $\mu \pm 2\sigma$ ) des offenen Regelkreises ist in Blau, die des geschlossenen in Türkis und die Messung in Rot dargestellt. Die graue Fläche stellt die Prädiktion des Modellfehlers für die aktuelle Messung dar.

welche dafür sorgt, dass nur ein Teil der Zustandsraumgleichungen korrigiert werden muss. Der hybride Modellansatz lautet somit

$$\mathbf{x}_{k+1} = \hat{\mathbf{f}}(\mathbf{x}_k, u_k) + \mathbf{B}\Delta\dot{\boldsymbol{\varphi}}_{k+1}(\boldsymbol{\varphi}_k, \dot{\boldsymbol{\varphi}}_k). \quad (4.65)$$

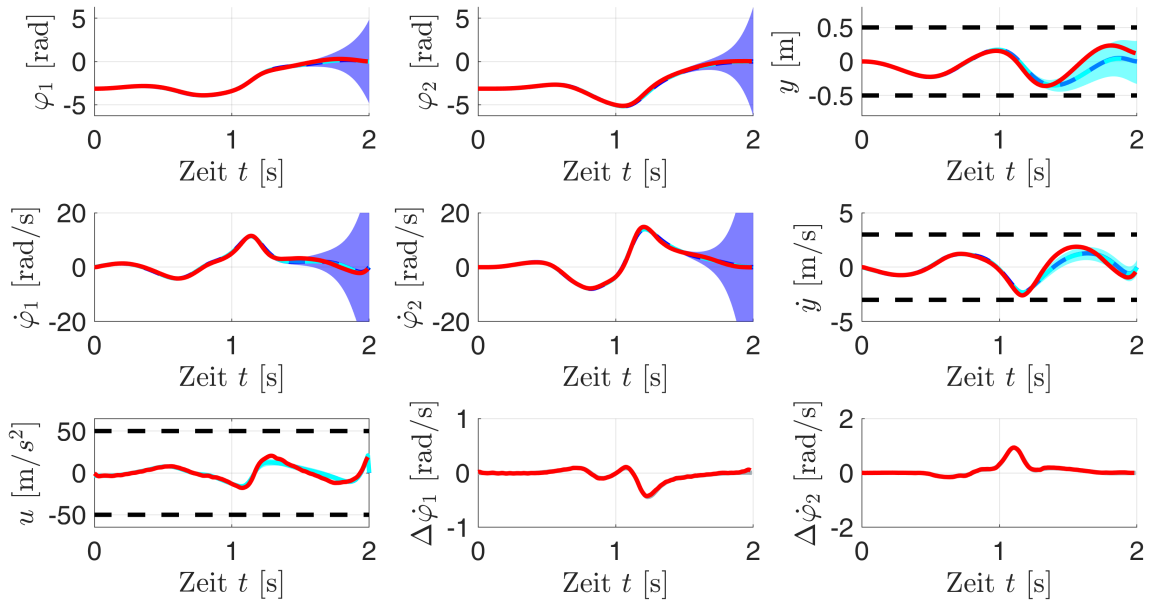
Das betrachtete Szenario beinhaltet eine zeitliche Diskretisierung von  $\Delta t = 0.01$  s mit einer fest vorgegebenen Aufschwingzeit von  $T = 2$  s. Das System startet aus dem Anfangszustand  $\mathbf{m}_I = [-\pi, -\pi, \mathbf{0}]^T$ ,  $\mathbf{S}_I = 10^{-3}\mathbf{I}_6$  und soll zum Zielzustand  $\mathbf{x}_G = \mathbf{0}$  überführt werden. Um die technischen Rahmenbedingungen mit einem ausreichend großen Sicherheitsabstand zu adressieren, werden die Nebenbedingungen mit

$$|s| \leq 0.5 \text{ m}, \quad |\dot{s}| \leq 3 \frac{\text{m}}{\text{s}}, \quad |u| = |\dot{s}| \leq 50 \frac{\text{m}}{\text{s}^2}, \quad \text{und} \quad P_x = 95\% \quad (4.66)$$

berücksichtigt. Die Gewichtungen des Optimalsteuerungsproblems (4.53) haben die Werte

$$\mathbf{W}_x = \text{diag}(100, 100, 250, \mathbf{0}), \quad W_u = 1, \quad w_t(k) = \frac{e^{25\Delta tk} - 1}{e^{25Hk} - 1}. \quad (4.67)$$

Abbildung 4.8 zeigt die zeitlichen Verläufe der 3. Lerniteration am Doppelpendelprüfstand. Die Unsicherheit über das Systemverhalten bei Verwendung der optimalen Steuerung (blau) steigt erwartungsgemäß mit zunehmender Zeit an. Gut erkennbar ist, dass die Hinzunahme des Reglers (türkis) zu einem instabilen Verhalten führt. Die Instabilität tritt bei etwa 1.6s auf, wobei an dieser Stelle die Winkelgeschwindigkeiten ihren maximalen Wert annehmen und sich die fehlenden Zentrifugalkräfte besonders stark bemerkbar machen. Aus dieser Perspektive betrachtet, ist die Instabilität daher nachvollziehbar und plausibel. Hier sei darauf hingewiesen, dass



**Abbildung 4.9** Ergebnisse der 5. bzw. letzten Lerniteration der hybriden Optimalsteuerung am Doppelpendel auf einem Wagen. Die Prädiktion ( $\mu \pm 2\sigma$ ) des offenen Regelkreises ist in Blau, die des geschlossenen in Türkis und die Messung in Rot dargestellt. Die graue Fläche stellt die Prädiktion des Modellfehlers für die aktuelle Messung dar.

der Riccati-Regler aufgrund seiner Herleitung bzgl. des hybriden Modells entlang der Trajektorie theoretisch stabil sein sollte. Bei einem herkömmlichen deterministischen Entwurf würde der/die Regelungstechniker\*in also fälschlicherweise von einem stabilen Systemverhalten ausgehen. Die probabilistische berechnete Instabilität kann daher nur das Resultat eines nicht vollständig bekannten realen Systems sein und ist für den/die Entwurfsingenieur\*in ein wichtiges Hilfsmittel, um das Verhalten am Prüfstand vor dem Experiment besser einschätzen zu können. Entsprechend der Stabilitätsprüfung in Algorithmus 1 wird am Prüfstand lediglich die Steuerung verwendet und der rote Verlauf gemessen. Anhand der Diagramme zum Modellfehler wird ersichtlich, dass der datengetriebene GP-Modellteil noch nicht in der Lage ist eine vollständige Korrektur vorzunehmen. Weitere Messdaten sind notwendig, um die hohe Unsicherheit weiter zu verringern, sodass eine weitere Iteration gestartet wird. Zwei weitere Iterationen sind erforderlich, bevor der Aufschwung der Pendelarme und die Stabilisierung am Prüfstand realisiert werden können. Abbildung 4.9 zeigt das finale Ergebnis: Die Prädiktion des geschlossenen Regelkreises weist nur eine sehr geringe Unsicherheit auf, welche insbesondere für die Winkelverläufe nicht sichtbar ist. Der/Die Regelungstechniker\*in kann somit gewährleisten, dass die Regelung am Prüfstand stabil funktionieren wird und sich das System entsprechend der Vorhersage verhalten wird. Die Messungen bestätigen dieses Verhalten, wobei die geforderten Rahmenbedingungen (schwarze gestrichelte Linien) für die Stellgröße und die Zustandsgrößen bzgl. des Wagens eingehalten werden. Anhand der Diagramme der Modellfehler wird zudem ersichtlich, dass das GP-Modell die vorab unbekannten Zentrifugalkräfte mit einer hohen Güte wiedergeben kann und die Korrektur somit zuverlässig funktioniert. Eine ausführliche Darstellung dieser Arbeiten kann in [18], [175], [176] gefunden werden.

Die hybride Optimalsteuerung bietet ein vielversprechendes Verfahren für die Trajektorien-Planung und -Realisierung von Systemen, die nur teilweise in Form von **unvollständigen Modellen** bekannt sind und damit Fehler enthalten. Ein wesentlicher Vorteil dieses Ansatzes besteht darin, dass **Unsicherheiten** über die vorhandenen Systemfehler explizit in die Planungsphase einbezogen werden. Dadurch wird nicht nur die Genauigkeit der Steuerung erhöht, sondern auch die **Sicherheit** bei der praktischen Erprobung des Gesamtsystems verbessert. Trotz dieser Vorteile ist jedoch zu beachten, dass die hybride Optimalsteuerung insgesamt mit einem **höheren Rechenaufwand** verbunden ist. Die umfassende Berücksichtigung der Unsicherheiten sowie die komplexen Berechnungen erfordern mehr Rechenressourcen und Zeit im Vergleich zu einfacheren Steuerungsverfahren.

#### 4.4.2 Hybride Zustandslinearisierung für eingangs-affine Systeme

Der Regelungsentwurf nach dem Verfahren der Zustandslinearisierung ist ein gängiger regelungstechnischer Ansatz, welcher nur unter besonderen Voraussetzungen angewendet werden kann [21]. Die Methode zielt darauf ab, ein nichtlineares System durch eine geeignete Wahl der Stellgröße als ein lineares System zu behandeln. Hierbei werden die vorhandenen Nichtlinearitäten im Idealfall vollständig kompensiert. Eine Linearisierung durch eine Taylorreihenentwicklung findet dabei nicht statt, sodass das System im gesamten Zustandsraum als linear betrachtet werden darf. Dies hat den großen Vorteil, dass die Methoden der linearen Regelungstechnik zum Entwurf und zur Analyse angewendet werden dürfen und entsprechend ein linearer Zustandsregler zur Stabilisierung und Sollwertfolge ausreichend ist und vergleichsweise einfach bestimmt werden kann. Damit die Methodik anwendbar ist, muss die Voraussetzung gelten, dass es sich um ein eingangsaffines System handelt, d. h. es gilt

$$\dot{\mathbf{x}} = \boldsymbol{\alpha}(\mathbf{x}) + \boldsymbol{\beta}(\mathbf{x})\mathbf{u}. \quad (4.68)$$

Um die Kompensation der Nichtlinearitäten zu erreichen, ist die passende Wahl der Stellgröße am realen System erforderlich. Die Stellgröße hängt hierbei von bestimmten Modelltermen ab, welche die Nichtlinearitäten des realen Systems genau genug wiedergeben müssen. Entsprechend der Grundidee, werden die Terme über die Stellgröße dem realen System so aufgeschaltet, sodass sie invers zur realen Systemdynamik wirken und sich bei einer gesamtheitlichen Betrachtung möglichst ideal kompensieren. Ein möglicher Regelungsansatz lautet dann

$$\mathbf{u}_c(\mathbf{x}) = \hat{\boldsymbol{\beta}}(\mathbf{x})^{-1}(-\hat{\boldsymbol{\alpha}}(\mathbf{x}) + \mathbf{K}(\mathbf{w} - \mathbf{x})), \quad (4.69)$$

wobei  $\hat{\boldsymbol{\alpha}}$  und  $\hat{\boldsymbol{\beta}}$  die modellbasierten Schätzungen der Nichtlinearitäten sind und ein Ansatz für einen P-Regler verwendet wird. Weist das Modell dahingehend Ungenauigkeiten auf, so wirken sich diese über das aufgestellte Regelgesetz direkt auf die Regelgüte aus. Unter Umständen werden durch das ungenaue Modell zusätzliche unerwünschte Nichtlinearitäten eingebracht, welche negative Effekte, wie Grenzyklen oder Instabilitäten, nach sich ziehen können. Das Ziel des hier vorgestellten Ansatz-

zes ist, eine hybride Zustandslinearisierung zu entwickeln, welche die angesprochenen Ungenauigkeiten auf der Basis von Messdaten und der Gauß-Prozess-Regression korrigiert. In diesem Szenario ist ein teilweise unvollständiges physikalisches Modell zulässig, auf dessen Grundlage eine ideale Kompensation der Zustandslinearisierung nicht funktioniert.

Um das Grundprinzip nachvollziehen zu können, ist die Betrachtung einer Differentialgleichung  $i \in \{1, \dots, n_x\}$  des Gesamtsystems aus Gleichung (4.68) ausreichend:

$$\dot{\mathbf{x}} = \beta_1(\mathbf{x})u_1 + \dots + \beta_{n_u}(\mathbf{x})u_{n_u} + \alpha(\mathbf{x}). \quad (4.70)$$

Zur besseren Übersicht wurde dabei der Index  $i$  weggelassen. Den Ausgangspunkt des Verfahrens bilden die  $n_d$  aufgenommenen Datenpaare  $(\dot{\mathbf{x}}, \mathbf{x}, \mathbf{u})$ , welche mit dem standardmäßigen Ansatz für das Messrauschen  $\epsilon_n \sim \mathcal{N}(0, \sigma_n^2)$  additiv verfälscht sind und in den nachfolgenden Matrizen und dem Ausgangsdatenvektor zusammengefasst werden:

$$\begin{aligned} \mathbf{X} &= [\mathbf{x}^{(1)}, \dots, \mathbf{x}^{(n_d)}] \in \mathbb{R}^{n_x \times n_d}, \\ \mathbf{U}_j &= \text{diag}(u_j^{(1)}, \dots, u_j^{(n_d)}) \in \mathbb{R}^{n_d \times n_d}, \quad j = 1, \dots, n_u, \\ \dot{\mathbf{X}} &= [\dot{\mathbf{x}}^{(1)}, \dots, \dot{\mathbf{x}}^{(n_d)}]^T \in \mathbb{R}^{n_x \times n_d}. \end{aligned} \quad (4.71)$$

Die gesuchten Funktionen werden als Zufallsvariablen, die jeweils von einem GP stammen, definiert

$$\begin{aligned} \alpha(\mathbf{x}) &\sim \mathcal{GP}(m_\alpha(\mathbf{x}), k_\alpha(\mathbf{x}, \mathbf{x}')), \\ \beta_j(\mathbf{x}) &\sim \mathcal{GP}(m_{\beta_j}(\mathbf{x}), k_{\beta_j}(\mathbf{x}, \mathbf{x}')), \quad j = 1, \dots, n_u, \end{aligned} \quad (4.72)$$

Ein möglicherweise vorhandenes Vorwissen kann hierbei über die Mittelwertfunktionen  $m_\alpha(\mathbf{x})$  und  $m_{\beta_j}(\mathbf{x})$  berücksichtigt werden. Für die Kovarianzfunktionen wird standardmäßig der SE-Kernel angesetzt. Es wird nun ein Zustand  $\mathbf{x}^*$  eingeführt, an welchem die Funktionen ausgewertet werden sollen. Zusammen mit den Ausgangsdaten, lässt sich durch die Definition als GPs für die Prior-Verteilung folgern, dass

$$\begin{bmatrix} \alpha(\mathbf{x}^*) \\ \beta_1(\mathbf{x}^*) \\ \vdots \\ \beta_{n_u}(\mathbf{x}^*) \\ \dot{\mathbf{X}} \end{bmatrix} \sim \mathcal{N} \left( \begin{bmatrix} m_\alpha(\mathbf{x}^*) \\ m_{\beta_1}(\mathbf{x}^*) \\ \vdots \\ m_{\beta_{n_u}}(\mathbf{x}^*) \\ \mathbf{m}_{\dot{\mathbf{X}}} \end{bmatrix}, \begin{bmatrix} k_\alpha^* & 0 & 0 & \dots & \mathbf{k}_\alpha^T \\ 0 & k_{\beta_1}^* & 0 & \dots & \mathbf{k}_{\beta_1}^T \mathbf{U}_1^T \\ \vdots & \vdots & \ddots & \dots & \vdots \\ 0 & 0 & \dots & k_{\beta_{n_u}}^* & \mathbf{k}_{\beta_{n_u}}^T \mathbf{U}_{n_u}^T \\ \mathbf{k}_\alpha & \mathbf{U}_1 \mathbf{k}_{\beta_1} & \dots & \mathbf{U}_{n_u} \mathbf{k}_{\beta_{n_u}} & \mathbf{K}_{\dot{\mathbf{X}}} \end{bmatrix} \right) \quad (4.73)$$

gilt, mit  $k_{(\cdot)}^*(\mathbf{x}^*) = k_{(\cdot)}(\mathbf{x}^*, \mathbf{x}^*)$ ,  $\mathbf{k}_{(\cdot)}(\mathbf{x}^*) = [k_{(\cdot)}(\mathbf{x}^*, \mathbf{x}^{(1)}), \dots, k_{(\cdot)}(\mathbf{x}^*, \mathbf{x}^{(n_d)})]^T \in \mathbb{R}^{n_d}$  und wobei die grundlegenden Regeln zur Addition und Produktbildung von GPs angewendet wurden [177]. Dadurch ergeben sich insbesondere für die Ausgangsdaten der Erwartungsvektor und die Kovarianzmatrix zu

$$\begin{aligned} \mathbf{m}_{\dot{\mathbf{X}}} &= \mathbf{m}_\alpha(\mathbf{X}) + \mathbf{U}_1 \mathbf{m}_{\beta_1}(\mathbf{X}) + \dots + \mathbf{U}_{n_u} \mathbf{m}_{\beta_{n_u}}(\mathbf{X}), \quad \text{und} \\ \mathbf{K}_{\dot{\mathbf{X}}} &= \mathbf{K}_\alpha + \mathbf{U}_1^T \mathbf{K}_{\beta_1} \mathbf{U}_1 + \dots + \mathbf{U}_{n_u}^T \mathbf{K}_{\beta_{n_u}} \mathbf{U}_{n_u} + \sigma_n^2 \mathbf{I}_{n_d}, \end{aligned} \quad (4.74)$$

mit  $\mathbf{m}_{(\cdot)}(\mathbf{X}) = [m_{(\cdot)}(\mathbf{x}^{(1)}), \dots, m_{(\cdot)}(\mathbf{x}^{(n_d)})]^T \in \mathbb{R}^{n_d}$  und  $\mathbf{K}_{(\cdot)} \in \mathbb{R}^{n_d \times n_d}$  mit den Elementen  $K_{(\cdot)}^{(i,j)} = k_{(\cdot)}(\mathbf{x}^{(i)}, \mathbf{x}^{(j)})$ ,  $i, j = 1, \dots, n_d$ . Auf der Grundlage der Prior-Verteilung werden die bedingten bzw. Posterior-Verteilungen nach

$$\begin{aligned} \alpha(\mathbf{x}^*) \mid \dot{\mathbf{X}} &\sim \mathcal{N}(\mu_\alpha(\mathbf{x}^*), \sigma_\alpha^2(\mathbf{x}^*)), \\ \mu_\alpha(\mathbf{x}^*) &= m_\alpha(\mathbf{x}^*) + \mathbf{k}_\alpha^T(\mathbf{x}^*) \mathbf{K}_{\dot{\mathbf{X}}}^{-1}(\dot{\mathbf{X}} - \mathbf{m}_{\dot{\mathbf{X}}}), \\ \sigma_\alpha^2(\mathbf{x}^*) &= k_\alpha^*(\mathbf{x}^*) - \mathbf{k}_\alpha^T(\mathbf{x}^*) \mathbf{K}_{\dot{\mathbf{X}}}^{-1} \mathbf{k}_\alpha(\mathbf{x}^*), \end{aligned} \quad (4.75)$$

$$\begin{aligned} \beta_j(\mathbf{x}^*) \mid \dot{\mathbf{X}} &\sim \mathcal{N}(\mu_{\beta_j}(\mathbf{x}^*), \sigma_{\beta_j}^2(\mathbf{x}^*)), \quad j = 1, \dots, n_u, \\ \mu_{\beta_j}(\mathbf{x}^*) &= m_{\beta_j}(\mathbf{x}^*) + \mathbf{k}_{\beta_j}^T(\mathbf{x}^*) \mathbf{U}_j^T \mathbf{K}_{\dot{\mathbf{X}}}^{-1}(\dot{\mathbf{X}} - \mathbf{m}_{\dot{\mathbf{X}}}), \\ \sigma_{\beta_j}^2(\mathbf{x}^*) &= k_{\beta_j}^*(\mathbf{x}^*) - \mathbf{k}_{\beta_j}^T(\mathbf{x}^*) \mathbf{U}_j^T \mathbf{K}_{\dot{\mathbf{X}}}^{-1} \mathbf{U}_j \mathbf{k}_{\beta_j}(\mathbf{x}^*) \end{aligned}$$

bestimmt. Es liegt somit eine probabilistische Schätzung in Form einer eindimensionalen Normalverteilung für jede der gesuchten Funktionen der Zustandslinearisierung vor, wobei die angesetzten Prior-Mittelwertfunktionen mit einbezogen werden und geeignet korrigiert werden. Die beschriebene Methode wird für alle  $n_x$  Differentialgleichungen (4.68) gleichermaßen durchgeführt, sodass eine Zusammenführung aller Posterior-Mittelwerte kurz über

$$\boldsymbol{\mu}_\alpha(\mathbf{x}) = \begin{bmatrix} \mu_{\alpha_1}(\mathbf{x}) \\ \vdots \\ \mu_{\alpha_{n_x}}(\mathbf{x}) \end{bmatrix} \in \mathbb{R}^{n_x}, \quad \boldsymbol{\mu}_\beta(\mathbf{x}) = \begin{bmatrix} \mu_{\beta_{1,1}}(\mathbf{x}) & \cdots & \mu_{\beta_{1,n_u}}(\mathbf{x}) \\ \vdots & \ddots & \vdots \\ \mu_{\beta_{n_x,1}}(\mathbf{x}) & \cdots & \mu_{\beta_{n_x,n_u}}(\mathbf{x}) \end{bmatrix} \in \mathbb{R}^{n_x \times n_u} \quad (4.76)$$

beschrieben wird. Diese Größen stellen auf der Basis des Vorwissens und der Datenlage die beste Schätzung für die gesuchten Funktionen dar. Aus diesem Grund wird hinsichtlich der Anwendung am realen System das Regelungsgesetz über diese Schätzungen mit

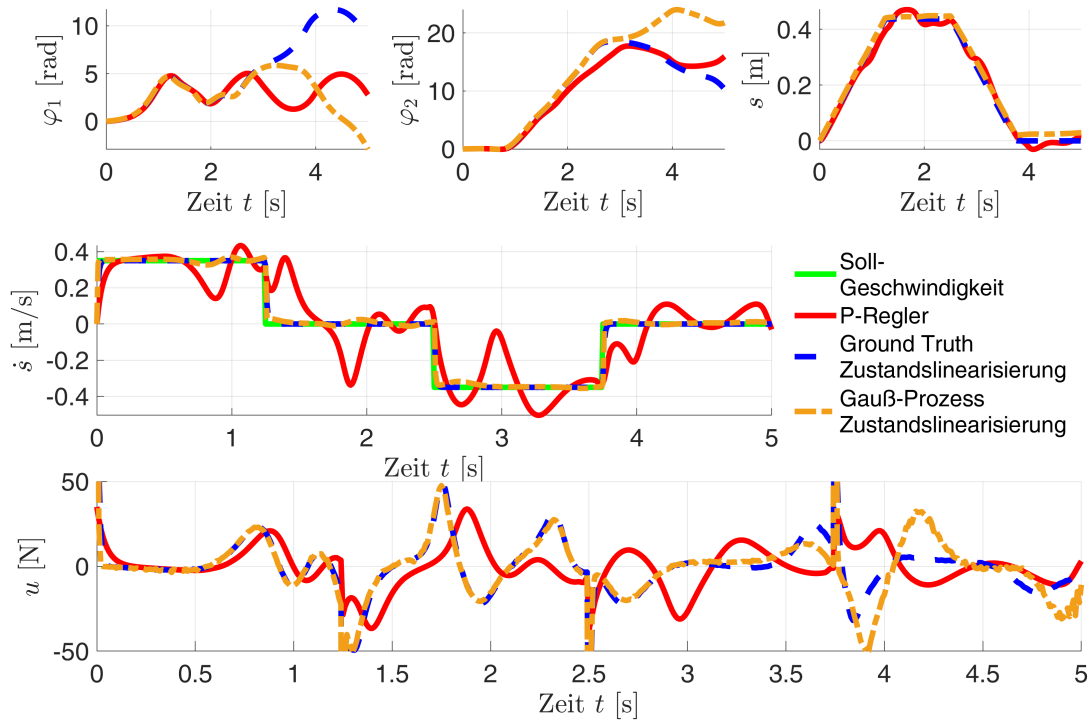
$$\mathbf{u}_c(\mathbf{x}) = \boldsymbol{\mu}_\beta(\mathbf{x})^{-1}(-\boldsymbol{\mu}_\alpha(\mathbf{x}) + \mathbf{K}(\mathbf{w} - \mathbf{x})) \quad (4.77)$$

definiert, sodass zu jedem Zeitpunkt eine eindeutige Stellgröße bekannt ist.

Als Nächstes wird das beschriebene Entwurfsverfahren an dem bereits eingeführten Doppelpendel auf einem Wagen (siehe Abbildung 4.7) illustriert. Diesmal wird nicht der Aufschwung der Pendelarme als Entwurfsziel definiert, sondern die genaue Einhaltung einer vorgegebenen Soll-Geschwindigkeit des Wagens. Für dieses Szenario wird die Stellgröße  $u = u_s$  nicht mehr als ideale Vorgabe der Wagenbeschleunigung angesehen (vgl. Gleichung (4.61)). Der neue Eingang  $u_F$  in das System ist die Kraft, welche auf den Wagen wirkt. Diese ergibt sich als Folge des Zusammenhangs

$$\begin{aligned} u_s = \frac{1}{m_1 + m_2 + m_0} & \left( (a_1 m_1 + l_1 m_2) \cos(\varphi_1) \ddot{\varphi}_1 + a_2 m_2 \cos(\varphi_2) \ddot{\varphi}_2 \right. \\ & \left. - (a_1 m_1 + l_1 m_2) \sin(\varphi_1) \dot{\varphi}_1^2 + a_2 m_2 \sin(\varphi_2) \dot{\varphi}_2^2 + u_F \right), \end{aligned} \quad (4.78)$$

welcher für das betrachtete Szenario in Gleichung (4.61) eingesetzt wird. Das Ziel der Regelung ist, die Wagenkraft so vorzugeben, dass die vorgegebene Soll-Geschwindigkeit



**Abbildung 4.10** Beispielhafte Anwendung der hybriden Zustandslinearisierung für eine Geschwindigkeitsregelung an einem Doppelpendel auf einem Wagen.

keit erreicht wird und sich die Bewegungen der Pendelarme nicht negativ auf das Ergebnis auswirken. Die Pendelarme fungieren somit als Störung, welche durch ein genaues Modell bzw. eine partielle Zustandslinearisierung kompensiert werden kann.

Abbildung 4.10 stellt verschiedene Regelungsansätze für das beschriebene Szenario gegenüber. Das mittlere Diagramm zeigt die zeitliche Entwicklung der Wagenbeschwindigkeit, wobei der Sollwert  $\dot{s}_{Soll}(t)$  in grün dargestellt ist und sich sprunghaft verändert. Ein herkömmlicher P-Regler (roter Verlauf) mit dem Stellgrößenansatz  $u_c(\dot{s}) = K(\dot{s}_{Soll}(t) - \dot{s}(t))$  ist nur bedingt in der Lage, die induzierte Störung der Pendelarme zu kompensieren und zeigt deutliche Abweichungen zum Sollwert auf. Herausfordernd ist hierbei, dass die Pendelarme in der oberen instabilen Ruhelage starten und sich damit eine sehr hohe Anfangsenergie im System befindet. Weiterführend werden die Daten, welche beim Versuch mit dem P-Regler aufgenommen wurden, nicht verworfen, sondern dienen als Grundlage für die hybride Zustandslinearisierung (vgl. Gleichung (4.71)). Die Anwendung des Verfahrens ergibt das zugehörige Regelungsgesetz (4.77), welches ein Modell der Pendelbewegung enthält und adäquat auf sie reagieren kann. Die orangefarbenen Verläufe verdeutlichen die Verbesserung zum einfachen P-Regler ohne Zustandslinearisierung. Zu beachten ist, dass der gleiche Verstärkungsfaktor  $K$  verwendet wurde. Als Referenz ist zudem das Ergebnis einer bestmöglichen Zustandslinearisierung gezeigt (Ground Truth, blau), welche auf einem exakten Modell basiert. Die gewählte Stellgröße in jedem Zeitschritt unterscheidet sich dabei kaum von der der gelernten GP-Regelung. Die hybride Zustandslinearisierung wurde auch an anderen Anwendungsbeispielen erfolgreich getestet und weiterführende Informationen dazu sind in [18] zu finden.



Die hybride Zustandslinearisierung wird für **eingangs-affine** Systeme mit separierten Nichtlinearitäten eingesetzt. A-Priori Schätzungen der Nichtlinearitäten werden auf der Basis von Zustandsmessungen und der **Gauß-Prozess-Regression** verbessert, um das zugrundeliegende Regelungsgesetz zu korrigieren und eine höhere Regelgüte zu erzielen. Zusätzlich kann über die Posterior-Varianzangabe zu den einzelnen Nichtlinearitäten dazu genutzt werden, um die Qualität der Korrektur abzuschätzen und Gebiete im Zustandsraum zu identifizieren, in denen noch wenig oder keine Zustandsmessungen vorliegen. In diesen Gebieten können entsprechend weitere Messungen vorgenommen werden, um die Qualität der Regelung weiter zu erhöhen. Da das Regelgesetz nur vom Erwartungsvektor und nicht von der Kovarianz abhängt, ist der Berechnungsaufwand im Vergleich zur hybriden Optimalsteuerung geringer.

### 4.4.3 Interaktiver Entwurf mit Bayesscher Optimierung

In den vorangegangenen Abschnitten 4.4.1 und 4.4.2 wurde indirekt davon ausgegangen, dass der Zustand des betrachteten System vollständig gemessen werden kann. Auf der Grundlage dieser Messdaten wurde anschließend eine Korrektur auf Modellebene vorgenommen, welche die Genauigkeit steigerte und einen verbesserten Steuerungs- und Regelungsentwurf ermöglichte. Insbesondere für mechanische Systeme, deren Zustandsgrößen sich maßgeblich durch einfach zu messende Positions- und Geschwindigkeitsgrößen zusammensetzen, ist die Voraussetzung eines vollständig bekannten Zustandsvektors in der Regel erfüllbar. Für bestimmte dynamische Systeme ist der Zustandsvektor jedoch nicht messbar, da es keine technisch realisierbare Sensorik gibt oder eine entsprechende Messeinrichtung zu aufwendig zu realisieren ist. In einigen Fällen ist überdies unklar welche Größen im Zustandsvektor enthalten sind, womit dieser undefiniert ist.

Ein mechatronisches Beispielsystem hierfür ist der Ultraschalldrahtbondprozess [178], wobei eine mechanische Schwingung im Ultraschallbereich, induziert durch eine Piezoaktorik, eingesetzt wird, um zwei metallische Partner miteinander zu verbinden. Der Prozess läuft auf atomarer Ebene ab und ist daher physikalisch schwer zu modellieren. Ebenso schwer gestaltet sich die Messung von bestimmten Größen, die sich während des Prozesses kontinuierlich verändern. Eine Ausnahme bildet das Prozessende, an dem die, für die Qualität wesentliche, Festigkeit der Verbindung durch eine zerstörende Messung identifiziert werden kann. In diesem Szenario kann zwar nicht der Zustandsvektor gemessen, dafür jedoch die Gütefunktion am Ende der Trajektorie ausgewertet werden. Außerdem liegt ein gewisses Expertenwissen in Form einer parametrisierten Steuerung vor, welche sich für das Erreichen eines möglichst hohen Gütefunktionswertes eignet. In diesem Szenario, welches auch für andere komplexe mechatronische Systeme vorliegen kann, bietet sich der Einsatz der Bayesschen Optimierung [179] für einen zielgerichteten Entwurf an. Dabei werden interaktiv Experimente direkt am realen System durchgeführt und anhand der erhaltenen Messdaten eine Korrektur auf der gedanklichen Ebene der Gütefunktion durch die Gauß-Prozess-Regression vorgenommen. Der Entwurf mit der Bayesschen Optimierung sucht auf diese Weise iterativ nach einer optimalen Parametrisierung

für den gewählten Steuerungsansatz. Da lediglich eine Auswertung der Gütefunktion notwendig ist, muss bei diesem Verfahren im Vergleich zur hybriden Optimalsteuerung oder Zustandslinearisierung keine Korrektur der Dynamikgleichungen vorgenommen werden. Dies stellt den wesentlichen Vorteil des Verfahrens dar. Eine vorab bekannte Schätzung der Gütefunktion kann jedoch neben der parametrisierten Ansatzfunktion als Vorwissen zur Effizienzsteigerung verwendet werden.

Das Verfahren geht formal von einer parametrisierten Steuerung mit

$$\mathbf{u} = \mathbf{s}(t; \boldsymbol{\theta}) \quad (4.79)$$

aus. Dabei ist wiederum  $\mathbf{u} \in \mathbb{R}^{n_u}$  die Stellgröße,  $t$  die Zeit und  $\boldsymbol{\theta} \in \mathbb{R}^{n_\theta}$  der Parametervektor. Zudem stellt  $\mathbf{s}$  den gewählten Ansatz dar. Das übergeordnete Ziel ist eine vorgegebene Gütefunktion  $J$  zu optimieren, woraus sich die optimale Parametrisierung

$$\boldsymbol{\theta}_* = \arg \min_{\boldsymbol{\theta}} J(\boldsymbol{\theta}) \quad (4.80)$$

ergibt. Der Grundgedanke der Bayesschen Optimierung ist, die Gütefunktion als parameterabhängige Zufallsvariable über einen Gauß-Prozess

$$J(\boldsymbol{\theta}) \sim \mathcal{GP}(m(\boldsymbol{\theta}), k(\boldsymbol{\theta}, \boldsymbol{\theta}')), \quad (4.81)$$

zu definieren, wobei dieser auf eine wählbare Mittelwert- und Kovarianzfunktion  $m(\boldsymbol{\theta})$  bzw.  $k(\boldsymbol{\theta}, \boldsymbol{\theta}')$  zurückgeführt wird. Für die Mittelwertfunktion bietet es sich an, eine Schätzung der Gütefunktion vorzusehen, wobei bspw. ein vorhandenes physikalisches Modell entsprechend der parametrisierten Steuerung ausgewertet wird. Durch die Messung der Gütefunktion stehen (Ein-/Ausgangs-)Daten der Form

$$\mathbf{D}_\theta = [\theta_1, \dots, \theta_{n_d}] \in \mathbb{R}^{n_\theta \times n_d}, \quad \mathbf{D}_J = [J_1, \dots, J_{n_d}]^T \in \mathbb{R}^{n_d} \quad (4.82)$$

zur Verfügung. Basierend auf den Daten ergibt sich die Posterior-Gleichung der Gütefunktion zu

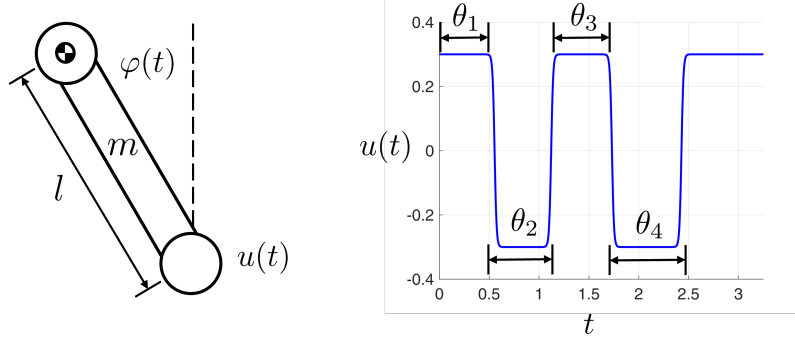
$$\begin{aligned} p(J(\boldsymbol{\theta}) \mid \mathbf{D}_J) &= \mathcal{N}(\mu(\boldsymbol{\theta}), \sigma^2(\boldsymbol{\theta})), \\ \mu(\boldsymbol{\theta}) &= m(\boldsymbol{\theta}) + \mathbf{k}_D^T(\boldsymbol{\theta}) \mathbf{K}^{-1} (\mathbf{D}_J - \mathbf{m}_D), \\ \sigma^2(\boldsymbol{\theta}) &= k(\boldsymbol{\theta}, \boldsymbol{\theta}) - \mathbf{k}_D^T(\boldsymbol{\theta}) \mathbf{K}^{-1} \mathbf{k}_D(\boldsymbol{\theta}), \end{aligned} \quad (4.83)$$

mit  $\mathbf{k}_D(\boldsymbol{\theta}) = [k(\boldsymbol{\theta}, \boldsymbol{\theta}_1), \dots, k(\boldsymbol{\theta}, \boldsymbol{\theta}_{n_d})]^T$ ,  $\mathbf{m}_D = [m(\boldsymbol{\theta}_1), \dots, m(\boldsymbol{\theta}_{n_d})]^T$  und symmetrischer, positiv definiter Gram-Matrix  $\mathbf{K} \in \mathbb{R}^{n_d \times n_d}$  mit den Elementen  $K_{i,j} = k(\boldsymbol{\theta}_i, \boldsymbol{\theta}_j)$ ,  $i, j = 1, \dots, n_d$ . Das iterative Vorgehen der BO ergibt sich danach durch eine kontinuierliche Erweiterung des Datensatzes

$$\mathbf{D}_\theta \leftarrow \{\mathbf{D}_\theta, \boldsymbol{\theta}_{n_d+1}\}, \quad \mathbf{D}_J \leftarrow \{\mathbf{D}_J, J_{n_d+1}\}, \quad (4.84)$$

wodurch die Schätzung in jeder Iteration weiter verbessert wird.

Bisher unklar ist, wie anhand des GPs die nächste Parametrisierung  $\boldsymbol{\theta}_{n_d+1}$  bestimmt bzw. nach welchem Kriterium das nächste Experiment am realen System festgelegt wird. Hierfür wird im Rahmen der BO eine sogenannte Akquisitionsfunk-



**Abbildung 4.11** Ein im Drehgelenk aktuiertes Pendel (links) und zugehörige parametrisierte Bang-Bang-Steuerung (rechts).

tion  $\alpha(\boldsymbol{\theta}) : \mathbb{R}^{n_\theta} \mapsto \mathbb{R}$ , welche von der aktuellen Schätzung bzgl. der Gütefunktion des GP abhängig ist, benutzt und ein unterlagertes Optimierungsproblem

$$\boldsymbol{\theta}_{n_d+1} = \arg \max_{\boldsymbol{\theta}} \alpha(\mu(\boldsymbol{\theta}), \sigma^2(\boldsymbol{\theta})) \quad (4.85)$$

automatisiert gelöst. Mittlerweile gibt es ein breites Spektrum von möglichen Funktionen, die je nach Anwendungsfall besser oder schlechter geeignet sind. Das bekannteste Kriterium lautet Expected Improvement (EI) [179] und folgt der Grundidee, die Größe der möglichen Verbesserung zu quantifizieren. Hierfür wird der Erwartungswert im Bezug auf die Dichte, die unter dem Grenzwert  $\xi_J$  liegt, mit

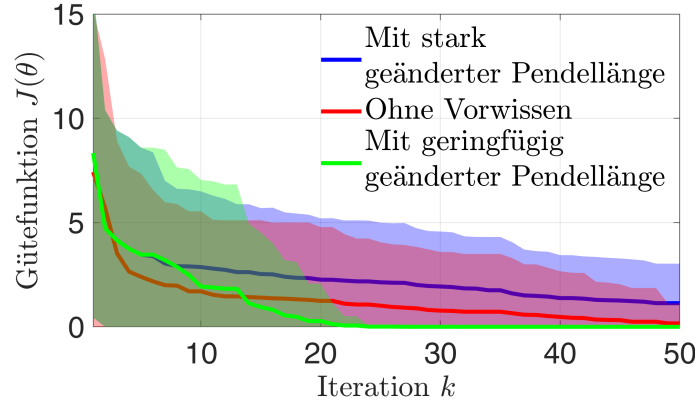
$$\begin{aligned} \alpha_{EI}(\boldsymbol{\theta}) &= \mathbb{E}[\max(0, \xi_J - J(\boldsymbol{\theta}))] \\ &= \sigma(\boldsymbol{\theta})(\gamma(\boldsymbol{\theta})\Phi(\gamma(\boldsymbol{\theta})) + \phi(\gamma(\boldsymbol{\theta}))), \\ \gamma(\boldsymbol{\theta}) &= (\xi_J - \mu(\boldsymbol{\theta}))/\sigma(\boldsymbol{\theta}) \end{aligned} \quad (4.86)$$

ausgewertet. Hierin ist  $\phi(\cdot)$  die Dichtefunktion und  $\Phi(\cdot)$  die Verteilungsfunktion einer Standardnormalverteilung. Zusammenfassend iteriert die Bayessche Optimierung zwischen dem Update des Posteriors (4.83) und der Durchführung des nächsten Experiments mit der optimierten Parametrisierung (4.85).

Abbildung 4.11 zeigt einen zu Anschauungszwecken konstruierten Anwendungsfall der Bayesschen Optimierung. Dabei soll ein Einfachpendel aus seiner unteren Ruhelage ( $\varphi(t=0) = -\pi, \dot{\varphi}(t=0) = 0$ ) aufgeschwungen werden und nach  $T = 3.1$  s den Zielwinkel  $\varphi(t=T) = 0$  rad erreichen. Da die Stellgröße mit  $u_{max} = 0.3$  N m auf einen vergleichsweise geringen Wert begrenzt ist, muss das Vorzeichen der Steuerung mehrfach wechseln, damit sich das Pendel (ähnlich zu einer Schaukel) aufschwingen kann. Die Pendeldynamik wird über

$$\ddot{\varphi} = \frac{g}{l} \sin(\varphi) + \frac{1}{m^2 l} u \quad (4.87)$$

beschrieben, wobei das Ground Truth System die Parameter:  $l = 0.3$  m,  $m = 0.5$  kg und  $g = 9.81 \frac{\text{m}}{\text{s}^2}$  besitzt. Für die Steuerung wird ein sogenannter Bang-Bang-Ansatz



**Abbildung 4.12** Anwendung der Bayesschen Optimierung zur Bestimmung einer optimalen Parametrisierung zum Aufschwung eines Einfachpendels. Dargestellt ist die Entwicklung der Gütefunktionswerte für verschiedene Priorfunktionen (Rot: Konstant Null, Blau: Abschätzung der Gütefunktion auf Basis eines Pendelmodells mit stark geänderter Länge, Grün: Identisch zu Blau, jedoch mit geringfügig geänderter Pendellänge). Die durchgezogenen Linien zeigen die mittlere Entwicklung und die schraffierten Flächen visualisieren die zweifache Standardabweichung.

[21] gewählt

$$u(t; \boldsymbol{\theta}) = u_{\max} \left( 1 + \sum_{i=1}^{n_{\theta}} \frac{2}{(1 + e^{(100(\sum_{j=1}^i \theta_j - t)))} (-1)^i \right), \quad (4.88)$$

wobei die Stellgröße zwischen den maximalen Werten hin und her wechselt (vgl. Abbildung 4.11 rechts). Der Parametervektor  $\boldsymbol{\theta}$  hat dabei die Dimension  $n_{\theta} = 4$  und definiert die Umschaltzeitpunkte der Steuerung. Für die Gütefunktion wird die quadratische Abweichung zum Zielzustand

$$J(\boldsymbol{\theta}) = \varphi(t = T)^2 \quad (4.89)$$

nach der Zeit  $T$  verwendet. Zusätzlich wird der Suchraum eingeschränkt durch  $0 \text{ s} \leq \theta_i \leq 1 \text{ s}$ ,  $i = 1, \dots, n_{\theta}$ .

Abbildung 4.12 zeigt die Ergebnisse der Bayesschen Optimierung für den Steuerungsentwurf. Auf der horizontalen Achse ist die Anzahl der Iterationen abgebildet und auf der Hochachse der Gütefunktionswert. Für einen Vergleich sind drei verschiedene Durchläufe mit unterschiedlichen Mittelwertfunktionen  $m(\boldsymbol{\theta})$  gezeigt. Bei den roten Durchläufen wurde  $m(\boldsymbol{\theta}) = 7$  für eine sogenannte pessimistische Schätzung verwendet. Bei den blauen und grünen Verläufen wurde die Gütefunktion mittels eines veränderten Pendelmodells (4.87) abgeschätzt und als Mittelwertfunktion verwendet. Gut zu erkennen ist, dass eine geringe Abweichung zwischen dem Modell für die Gütefunktionsschätzung und dem Ground Truth System einen positiven Effekt auf die Konvergenz und die Identifikation der optimalen Parametrisierung hat. Nach ungefähr 20 Iterationen wird der optimale Wert von 0 zuverlässig und robust erreicht, was sich durch den stetig abfallenden Mittelwert und die verschwindende Varianz, welche durch 25 separate Durchläufe identifiziert worden ist, zeigt. Ohne die Einbringung von Vorwissen ist die Konvergenzgeschwindigkeit reduziert und die

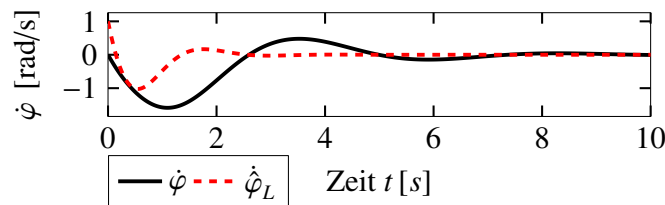
optimale Lösung wird im Durchschnitt erst nach 50 Iterationen erreicht. Die Einbringung von Vorwissen kann unter Umständen jedoch auch einen negativen Effekt auf die Suche haben (blauer Verlauf). Dabei weicht das zugrundeliegende Modell zu stark von der Realität ab, sodass die Bayessche Optimierung ungeeignete Orte im Suchraum der Parameter auswertet, um den Fehler auf Ebene der Gütefunktion zu korrigieren. Dies geht mit einer deutlich reduzierten Konvergenzgeschwindigkeit einher und führt auf einen ineffizienten Steuerungsentwurf [180].

Die Bayessche Optimierung stellt ein **globales Optimierungsverfahren** dar. Sie zeichnet sich durch ihre Flexibilität aus und ist eine hervorragende Wahl, wenn die **Zustände** eines Systems entweder **unbekannt**, undefiniert oder aufgrund aufwendiger Messungen bzw. nicht realisierbarer Messtechnik nicht direkt messbar sind. In solchen Szenarien wird die Korrektur nicht auf der Ebene der Systemzustände selbst vorgenommen, sondern durch eine Anpassung der Gütefunktion. Dadurch ist lediglich eine **Auswertung der Gütefunktion** notwendig, was den Optimierungsprozess erheblich vereinfacht. Es wird eine **parametrisierte Steuerung** benötigt, welche auf der Basis von Expertenwissen ausgewählt werden sollte. Die **Anzahl der zu optimierenden Parameter** sollte dabei im Bereich von weniger als 20 liegen, damit die Komplexität des Suchraums in einem überschaubaren Rahmen gehalten wird und gleichzeitig eine effiziente Optimierung vorgenommen werden kann. Durch den Einsatz der Bayesschen Optimierung kann die Inbetriebnahme technischer Systeme mit schwieriger Modellbildung unterstützt werden.



## 5 Beobachterentwurf

Sind nicht alle Zustandsgrößen messbar, auf deren Kenntnis die Regelung basiert, ist der Entwurf eines Beobachters erforderlich. Dieser schätzt die nicht messbaren Zustände üblicherweise modellbasiert. Aufgrund von Vereinfachungen in der Modellbildung und nicht modellierten Phänomenen (vgl. Kapitel 3) kann das genutzte Modell jedoch größere Abweichungen zur zu schätzenden Strecke aufweisen, sodass fehlerhafte oder unzureichend genaue Schätzwerte zu erwarten sind. Abbildung 5.1 illustriert diese Herausforderung durch einen Zustandsbeobachter, der die Winkelgeschwindigkeit eines Massenschwingers  $\dot{\varphi}_L$  aufgrund einer nicht modellierten sinusförmigen Kraft inkorrekt schätzt. Die sogenannte Modell-Realitäts-Lücke, welche durch solche Modellungenauigkeiten, aber auch externe Störungen ausgelöst wird, kann sogar Instabilität oder Divergenz eines Beobachters verursachen und folglich zu schwerwiegenden Auswirkungen auf die Zustandsregelung führen. Dieses Kapitel adressiert diesen Umstand, indem Methoden zur gleichzeitigen Schätzung von Zuständen und Modellungenauigkeiten in Abschnitt 5.2 sowie Koopman-basierte Techniken in Abschnitt 5.4 vorgestellt werden. Ferner werden Maßnahmen zur langfristigen Adaption des Modells in Abschnitt 5.3 präsentiert, welche innerhalb eines Zustandsbeobachters genutzt werden können und so ebenfalls zu einer Verminderung der Modell-Realitäts-Lücke beitragen.



**Abbildung 5.1** Modellungenauigkeiten führen im modellbasierten Beobachterentwurf zu ungenauen Schätzungen, am Beispiel des Massenschwingers

### 5.1 Grundlagen zum Beobachterentwurf

Nicht messbare Größen, wie Parameter oder Systemzustände, können mithilfe eines Beobachters rekonstruiert werden, wenn bestimmte Voraussetzungen erfüllt sind. Weist ein lineares, zeitinvariantes System die Eigenschaft *Beobachtbarkeit* auf, so ist es möglich, einen beliebigen Anfangszustand  $\mathbf{x}_0 \in \mathbb{R}^n$  in endlicher Zeit aus der Kenntnis der Ein- und Ausgangsgrößen  $\mathbf{u} \in \mathbb{R}^p$  bzw.  $\mathbf{y} \in \mathbb{R}^m$  zu schätzen. Diese Eigenschaft kann beispielsweise mit dem Kriterium nach Kalman oder Hautus überprüft werden [24]. Für nichtlineare Systeme ist diese Eigenschaft schwieriger zu definieren, daher wird häufig stattdessen die *schwache Beobachtbarkeit* genutzt, indem innerhalb einer Umgebung  $U = \{\mathbf{x}_0 \mid \|\mathbf{x}_0 - \mathbf{x}_p\| \} < \delta$  eines Punktes  $\mathbf{x}_p$

überprüft wird, ob der Anfangszustand  $\mathbf{x}_0$  aus der Kenntnis von Eingang  $\mathbf{u}$  und Ausgang  $\mathbf{y}$  eindeutig in endlicher Zeit rekonstruiert werden kann [21]. Ist dies für beliebige Punkte  $\mathbf{x}_p$  erfüllt, so heißt das nichtlineare System schwach beobachtbar. Um die Qualität eines Zustandsbeobachters zu beurteilen, wird zudem ein Schätzfehler  $\mathbf{e} = \hat{\mathbf{x}} - \mathbf{x}$  definiert, der die Abweichung des geschätzten Zustands  $\hat{\mathbf{x}}$  vom tatsächlichen Zustand  $\mathbf{x}$  bestimmt. Strebt dieser Fehler in überschaubarer Zeit gegen Null, ist eine hohe *Schätzgüte* gegeben [21], [24]. Für die Anwendung der Methoden dieses Kapitels wird daher vorausgesetzt, dass die betrachteten Systeme beobachtbar sind. Bevor ein Verfahren zur gleichzeitigen Schätzung von Zuständen und Modellungenauigkeiten vorgestellt wird, werden zunächst einige bekannte Zustandsbeobachter wiederholt und bzgl. ihres Einsatzes eingeordnet.

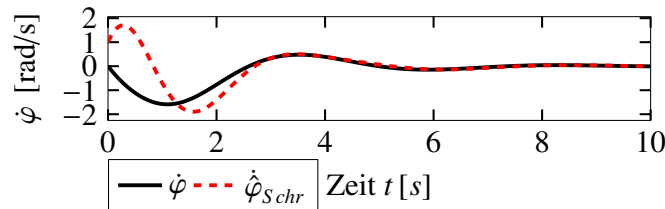
**Luenberger-Beobachter** Der Luenberger-Beobachter ist ein deterministischer Zustandsbeobachter für lineare Systeme [181]. Aufgrund seiner Struktur kann dieser Beobachter Anfangsstörungen  $\hat{\mathbf{x}}_0 \neq \mathbf{x}_0$  kompensieren, nicht jedoch beliebig auftretende Ungenauigkeiten [24]. Dennoch ist der Luenberger-Beobachter populär, da er bei Kenntnis eines genauen Streckenmodells mit wenig Aufwand über eine Polvorgabe auszulegen ist. So wird er beispielsweise für den Betrieb des Golfroboters mithilfe einer Gain-Scheduling-Strategie genutzt (vgl. Abschnitt 6.1), um die Zustände trotz nichtlinearer Systemdynamik schätzen zu können. Existieren unbekannte, aber isolierte Nichtlinearitäten in der Systemdynamik, ermöglicht der hybride Ansatz des *lernfähigen Beobachters* [182] eine Zustandsschätzung, da die Nichtlinearitäten durch ein neuronales Netz approximiert werden, während die bekannte lineare Dynamikstruktur weiterhin durch die traditionelle Beobachterstruktur adressiert wird. Eine Stabilitätsanalyse und die Abschätzung von Fehlerschranken gewährleisten dabei die Konvergenz des Netzes und folglich die Funktionsweise dieses Beobachters. Der lernfähige Beobachter kann in vielen Situationen eine sehr hilfreiche Maßnahme sein, wie die Abbildung 5.2 zeigt. Diese stellt anhand des zu Beginn eingeführten Beispiels des Massenschwingers die erfolgreiche Schätzung der Geschwindigkeit  $\dot{\varphi}_{Schr}$  durch den Beobachter nach Schröder dar, obwohl die Modellungenauigkeit weiterhin existiert. Allerdings weist die Nutzung eines neuronalen Netzes innerhalb eines Beobachters Nachteile wie z. B. die mangelnde physikalische Interpretierbarkeit aufgrund der Black-Box-Struktur auf, die bereits im Unterabschnitt 3.3.1 dargelegt worden sind.

**Kalman-Filter** Das Kalman-Filter unterscheidet sich zum Luenberger-Beobachter durch eine stochastische Perspektive [24], [183]. Es modelliert den zu schätzenden Systemzustand als Erwartungswert einer zugrunde liegenden Gauß-Verteilung und berücksichtigt mögliche Unsicherheiten wie beispielsweise Messrauschen durch Kovarianzen (vgl. [183]). Erweiterungen des Kalman-Filters zur Schätzung nichtlinearer Systemdynamiken bilden das Extended-Kalman-Filter (EKF), welches anhand von Jacobimatrizen eine linearisierte Version des Kalman-Filters formuliert, und das Unscented-Kalman-Filter (UKF), das Erwartungswert und Kovarianz der zugrunde liegenden Wahrscheinlichkeitsverteilung approximiert [184]. Durch ihre stochastische Perspektive werden diese Filter häufig als Zustandsbeobachter eingesetzt, um kleinere Modellungenauigkeiten durch die Kovarianzen zu kompensieren und eine zufriedenstellende Schätzung zu ermöglichen. Bestehen allerdings größere Modellab-



weichungen, ist dies nur noch eingeschränkt sichergestellt. Hybride Ansätze adressieren diesen Umstand, indem z. B. die Kovarianzen mittels Bayesscher Optimierung geschickt initialisiert werden [185], [186] oder ein EKF die Modellungenauigkeiten basierend auf einer Linearkombination aus radialen Basisfunktionen mit kompakten Träger schätzt [187]. Jene Methoden bieten u. U. eine temporäre Abhilfe, geben jedoch keine physikalisch wertvolle parametrische Darstellung der Ungenauigkeiten und adaptieren das Modell nicht langfristig. Dies ist jedoch wünschenswert, um eine sich sukzessiv verbessernde Modellgüte zu erhalten, welche für einen modellbasierten Beobachter erforderlich ist.

**Sliding-Mode-Beobachter** Der Sliding-Mode-Beobachter (SMO) stellt einen Vertreter der robusten Zustandsbeobachter dar. Im Kontrast zu den vorherigen Beobachterttypen erlaubt dieser eine dynamische Schätzung von Modellungenauigkeiten zur Laufzeit. Dies gelingt durch die Verwendung von unstetigen Schaltfunktionen, z. B.  $\nu_i(e_y) = p_i \cdot \text{sign}(e_y)$ , welche vom Vorzeichen des Ausgangsfehlers  $e_y$  und dem Parameter  $p_i$  mit  $i = 1, \dots, n$  beeinflusst werden [188], [189], [190]. Aufgrund der Beobachterstruktur entspricht die  $n$ -te Schaltfunktion  $\nu_n(e_y)$  dem Modellfehler. Wegen dieser Eigenschaft wird der SMO besonders als Störbeobachter bzw. in der Störidentifikation eingesetzt, indem diese Schaltfunktion mittels eines Tiefpassfilters identifiziert wird [188], [189], [190]. Zudem erlaubt er eine robuste Schätzung mit asymptotischer Konvergenz, sobald er gut parametrisiert ist. Allerdings erfordert eine gute Parametrierung die Kenntnis einer oberen Schranke der Modellungenauigkeiten bzw. Störungen, welche mitunter nicht einfach zu bestimmen ist [188], [189], [190].



**Abbildung 5.2** Korrekte Schätzung mittels des lernfähigen Beobachters nach [182] am Beispiel des Massenschwingers

## 5.2 Joint Estimation

Zu Beginn dieses Kapitels ist bereits die Herausforderung von Modellungenauigkeiten im Kontext modellbasierter Beobachter thematisiert worden. Bei größeren Modellabweichungen besteht aufgrund dieser das Risiko einer unzuverlässigen Zustandsschätzung. Diese resultiert u. U. in einer verringerten Regelgüte oder wirkt sich auf die Stabilität des Beobachters aus. Klassischerweise werden solche Unsicherheiten in der Regelungstechnik durch robuste Beobachter, wie den Sliding-Mode-Beobachter, oder stochastisch basierte Filter adressiert. Letztere berücksichtigen beispielsweise ein gewisses Maß an Unsicherheit durch die Modellierung des Zustands mithilfe von Kovarianzen. Im Kontrast zu diesen Vorgehen der Kompensation von Modellungenauigkeiten erlaubt die nachfolgend vorgestellte Methode des

augmentierten Beobachters eine simultane Schätzung und Identifikation der Modellungenauigkeiten mit den Zuständen.

Aus der Zustands- und Parameterschätzung ist die gleichzeitige Schätzung mehrerer unbekannter, u. U. dynamischer Größen bereits bekannt. Der Begriff *Joint Estimation*<sup>1</sup> bezeichnet hierbei eine simultane Schätzung von Systemzuständen und physikalischen Parametern mittels eines augmentierten Zustands [191], [192], [193], [194]. Dieser erweiterte Zustandsvektor<sup>2</sup>  $\hat{\mathbf{x}} \in \mathbb{R}^{n+n_\theta}$  definiert sich durch die Systemzustände  $\hat{\mathbf{x}} \in \mathbb{R}^n$  und die Parameter des Systems  $\hat{\boldsymbol{\theta}} \in \mathbb{R}^{n_\theta}$ . Seine Dynamik wird über das folgende Modell abgebildet:

$$\begin{aligned} \dot{\hat{\mathbf{x}}} &= \begin{bmatrix} \dot{\hat{\mathbf{x}}} \\ \dot{\hat{\boldsymbol{\theta}}} \end{bmatrix} = \begin{bmatrix} \mathbf{f}(\hat{\mathbf{x}}, \mathbf{u}, \boldsymbol{\theta}) \\ \mathbf{0} \end{bmatrix}, \\ \hat{\mathbf{y}} &= \mathbf{h}(\hat{\mathbf{x}}, \mathbf{u}). \end{aligned} \quad (5.1)$$

In der ersten Zeile des Zustandsraummodells (5.1) ist die Dynamik  $\mathbf{f}$  des betrachteten Systems vertreten, während für die Parameter  $\hat{\boldsymbol{\theta}}$  ein zeitinvariantes Verhalten angenommen wird. Die Ausgangsgleichung  $\mathbf{h}$  wird weiterhin am nicht erweiterten Systemzustand  $\hat{\mathbf{x}}$  ausgewertet, da die physikalischen Parameter in der Regel nicht zu messen sind.

Soll dieses Konzept auf die Schätzung von Modellungenauigkeiten übertragen werden, ist eine Anpassung notwendig. Eine populäre Strategie aus der linearen Regression besteht darin, Linearkombination aus geeigneten Ansatzfunktionen zur Approximation von nichtlinearen Zusammenhängen zu nutzen (vgl. das Verfahren SINDy im Abschnitt 3.2 und [10], [11]). Daher wird eine Modellungenauigkeit  $g \in \mathbb{R}$  als Linearkombination aus den Funktionen  $\psi_i : \mathbb{R}^n \times \mathbb{R}^p \mapsto \mathbb{R}$  mit  $i = 1, \dots, n_\theta$  durch

$$g(\mathbf{x}, \mathbf{u}) \approx \sum_{i=1}^{n_\theta} \theta_i \cdot \psi_i(\mathbf{x}, \mathbf{u}) = \boldsymbol{\theta}^T \boldsymbol{\Psi}(\mathbf{x}, \mathbf{u}) \quad (5.2)$$

angenähert. Im Verlauf des weiteren Kapitels werden ausschließlich eingangsaffine Systeme mit relativem Grad  $\delta = n$  betrachtet, die folglich beobachtbar sind [21]. Des Weiteren wird angenommen, dass das jeweilige betrachtete System in nichtlinearer Regelungsnormalform vorliegt und die Wirkung der Modellungenauigkeit additiv erfolgt. Ist demnach das folgende System

$$\begin{aligned} \dot{\mathbf{x}} &= \begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \vdots \\ \dot{x}_n \end{bmatrix} = \begin{bmatrix} x_2 \\ x_3 \\ \vdots \\ \underbrace{a(\mathbf{x}) + b(\mathbf{x}, \mathbf{u}) + g(\mathbf{x}, \mathbf{u})}_{=: \mathbf{f}(\mathbf{x}, \mathbf{u})} \end{bmatrix} \\ y &= x_1. \end{aligned} \quad (5.3)$$

mit der Modellungenauigkeit  $g$  gegeben, so lässt sich unter der Annahme in Gl. (5.2)

<sup>1</sup>Dieser Begriff lässt sich ins Deutsche mit *gemeinsamer* oder *gleichzeitiger Schätzung* übersetzen.

Im Folgenden wird jedoch auch weiterhin der englische Fachbegriff genutzt.

<sup>2</sup>Der Index  $\hat{\bullet}$  stellt eine geschätzte Größe dar.

das augmentierte Modell zur Approximation der Ungenauigkeit durch

$$\begin{aligned} \dot{\tilde{\mathbf{x}}} = \begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \vdots \\ \dot{x}_n \\ \dot{\boldsymbol{\theta}} \end{bmatrix} &\approx \underbrace{\begin{bmatrix} x_2 \\ x_3 \\ \vdots \\ a(\mathbf{x}) + b(\mathbf{x}, \mathbf{u}) + \boldsymbol{\theta}^T \boldsymbol{\Psi}(\mathbf{x}, \mathbf{u}) \\ \mathbf{0} \end{bmatrix}}_{=: \tilde{\mathbf{f}}(\tilde{\mathbf{x}}, \mathbf{u})}, \\ y &= x_1 \end{aligned} \quad (5.4)$$

formulieren [17]. Hierbei beschreibt  $\tilde{\mathbf{x}}$  wiederum den augmentierten Zustand, der sowohl den Systemzustand  $\mathbf{x}$  als auch die Parameter  $\boldsymbol{\theta}$  der Linearkombination (5.2) umfasst. Die eingangsaffine Dynamik wird durch  $a(\mathbf{x})$  und  $b(\mathbf{x}, \mathbf{u})$  dargestellt, während für die Approximation der Modellungenauigkeit  $g$  erneut eine zeitinvariante Parameterdynamik vorausgesetzt wird. Ferner muss eine Bibliothek  $\boldsymbol{\Psi} \in \mathbb{R}^{n_\theta}$  festgelegt werden. Da grundlegendes Wissen über die Systemordnung sowie die Systemzustände wegen des bestehenden Teilmodells vorhanden ist, lautet eine Minimalanforderung an die Bibliothek folgendermaßen:

$$\boldsymbol{\Psi}_0(\mathbf{x}, \mathbf{u}) = [1, x_1, \dots, x_n, u_1, \dots, u_p]^T. \quad (5.5)$$

In der Dissertation [17] wurde experimentell gezeigt, dass eine sinnvoll gewählte Bibliothek mindestens die Zustände, die Eingänge sowie Konstanten enthalten sollte. In der Regel existiert jedoch Vorwissen bzw. Erfahrungswissen bzgl. der beobachteten Modellungenauigkeit, welches als Hypothesen in Form von physikalisch motivierten Funktionen  $\psi_i$  in der Bibliothek berücksichtigt werden kann. Beispielsweise handelt es sich bei der Ungenauigkeit um unerwünschte Vibrationen oder oszillierendes Verhalten, sodass trigonometrische Funktionen eine sinnvolle Wahl für Bibliotheksfunktionen darstellen. Um zusätzlich eine physikalisch-technische Interpretierbarkeit<sup>3</sup> zu gewährleisten, ist eine gewisse Überschaubarkeit der Funktionen zur Charakterisierung der Modellungenauigkeit im Kontrast zur Superposition vieler Terme vorzuziehen. Im Sinne des Prinzips *Occam's Razor*<sup>4</sup> reichen daher eine geringe Menge von Funktionen aus, um eine interpretierbare und verlässliche Identifikation der Modellungenauigkeit zu erhalten [28]. Diese Begrenzung lässt sich mit dem Konzept *Sparsity* adressieren, welches bereits im Verfahren SINDy im Abschnitt 3.2 angesprochen worden ist. So wird die Anforderung, dass nur eine wenige Bibliotheksterme benötigt werden, durch die Dünnbesetztheit des Parametervektors ausgedrückt. Dessen Anzahl der Nichtnullelemente kann mittels der  $\ell_0$ -Norm<sup>5</sup>

<sup>3</sup>Dies meint eine konkrete Zuordnung von parametrischen Termen zu physikalischen Wirkprinzipien, vgl. die Definition in der Dissertation [17].

<sup>4</sup>Dieses geht auf den Philosophen Wilhelm von Ockham (1288-1347) zurück, welches postuliert, dass aus vielen Möglichkeiten meist die einfachste Lösung (oder Erklärung) mit hoher Wahrscheinlichkeit die beste Option darstellt. Dieses heuristische Prinzip wird vielfach in der Modellbildung verschiedener Disziplinen und im ML-Kontext zur Modellauswahl berücksichtigt [28].

<sup>5</sup>Diese ist im mathematischen Sinne keine Norm, sondern eine Halb- oder Pseudonorm.

definiert werden [195]:

$$\|\boldsymbol{\theta}\|_0 = \#\{i|\theta_i \neq 0\} \leq n_{\boldsymbol{\theta},act} \ll n_{\boldsymbol{\theta}}. \quad (5.6)$$

In Gl. (5.6) wird die Anzahl der Nichtnullelemente auf maximal  $n_{\boldsymbol{\theta},act}$  festgelegt, welche wesentlich kleiner als die Dimension des Parametervektors  $n_{\boldsymbol{\theta}}$  sein soll. Nun kann das augmentierte Modell (5.4) in einen Zustandsbeobachter eingesetzt werden, beispielsweise in ein UKF. Allerdings ist die Anforderung in Gl. (5.6) bisher noch nicht umgesetzt. Der nachfolgende Abschnitt erläutert daher, wie diese im Prädiktor-Korrektor-Schema eines UKFs integriert werden kann.

### 5.2.1 Augmentierter Beobachterentwurf

Durch das zuvor vorgestellte Konzept der Joint Estimation ist es möglich, einen Zustandsbeobachter so zu augmentieren, dass er mithilfe der dünnbesetzten Parameter einer Linearkombination die Modellungenauigkeiten parallel zu den Zuständen schätzt. Aufbauend auf Vorarbeiten [194] wird der Entwurf anhand eines UKFs durchgeführt. Konkret wird die numerisch stabilere Variante des Square-Root-Unscented-Kalman-Filters (SRUKFs) gewählt, welche auf [192] zurückgeht. Die grundlegenden Eigenschaften des Filters, d. h. seine stochastische Struktur sowie sein Prädiktor-Korrektor-Schema, sollen jedoch erhalten bleiben. Ausgehend von der Minimierung des Schätzfehlers (vgl. [127], [196]) wird daher das Konzept der Dünnbesetztheit mittels zwei verschiedener Umsetzungen in diese Filterstruktur eingebunden und berücksichtigt, welche erstmals in der Dissertation [17] sowie in den Veröffentlichungen [197], [198] vorgestellt worden sind.

**Dünnbesetztheit mittels fester Schranke** Die erste Umsetzung basiert auf Methoden des Compressed Sensing [199], [200], [201]. Diese Disziplin befasst sich mit hoch dimensional Signalen  $\mathbf{s} \in \mathbb{R}^n$ , welche durch eine geeignete Basiswahl  $\boldsymbol{\Psi} \in \mathbb{R}^{n \times n}$  mittels einiger weniger Basisfunktionen  $\psi_i$  mit  $i = 1, \dots, n$  komprimiert dargestellt werden können:

$$\mathbf{s} = \boldsymbol{\Psi}\boldsymbol{\theta} = \sum_i \theta_i \psi_i. \quad (5.7)$$

Die Approximation kann dabei durch die Wahl von orthonormalen Basisfunktionen gewährleistet werden [195]. Da orthonormale Basen eine limitierte Wahl für die Erfassung von Signalen darstellen und das Signal zunächst aufwendig gemessen und anschließend komprimiert werden muss, adressieren die Methoden im Compressed Sensing diese Nachteile, indem beliebige Projektionen betrachtet werden und die Extraktion des Signals direkt aus der Messung  $\mathbf{y} \in \mathbb{R}^m$  erfolgt [199], [200]. Daher gilt für ein komprimiertes Signal Folgendes:

$$\mathbf{y} = \mathbf{H}\boldsymbol{\Psi}\boldsymbol{\theta} = \mathbf{H}\mathbf{s}. \quad (5.8)$$

Das komprimierte Signal lässt sich dabei durch die *Messmatrix*<sup>6</sup>  $\mathbf{H} \in \mathbb{R}^{m \times n}$  messen, da  $\mathbf{y} = \mathbf{H}\boldsymbol{\Psi}\boldsymbol{\theta}$  gilt. Obwohl das Gleichungssystem  $\mathbf{y} = \mathbf{H}\mathbf{s}$  aufgrund von  $m < n$

---

<sup>6</sup>In der Regelungstechnik ist damit die Ausgangsmatrix  $\mathbf{C}$  bzw. die Jacobimatrix der Messfunktion  $\mathbf{h}$  gemeint.

in der Regel unterbestimmt ist, kann durch die Kenntnis der Basis  $\Psi$  sowie des Wissens, dass der Vektor  $\theta$  dünnbesetzt ist, das Signal  $\mathbf{s}$  dennoch rekonstruiert werden. Durch Umformulierung des Signals in den Gleichungen (5.7) und (5.8) lässt sich die Rekonstruktion in das folgende  $\ell_0$ -Problem übersetzen:

$$\hat{\theta} = \arg \min_{\theta} \|\theta\|_0, \quad \text{sodass} \quad \mathbf{y} = \mathbf{H}\Psi\theta. \quad (5.9)$$

Allerdings ist das Optimierungsproblem (5.9) nicht konvex und nur durch Kombinatorik bzw. eine *Brute-Force-Suche* zu lösen<sup>7</sup>. Diese Eigenschaft der Nicht-Konvexität wird für eine zweidimensionale Variable anhand der Abbildung 5.3 illustriert, welche die durch verschiedene  $p$ -Normen resultierenden Regionen darstellt und für  $p < 1$  nicht konvexe Gebiete aufzeigt. Liegen genügend Messungen vor und ist die Messmatrix  $\mathbf{H}$  annähernd inkohärent zur Basis  $\Psi$ , d. h. die Zeilen der Messmatrix sind unabhängig von den Spalten der Basis, kann das Problem (5.9) jedoch äquivalent in der  $\ell_1$ -Norm formuliert werden [14], [28], [195], [199], [200], sodass eine vorteilhafte, konvexe Gestalt entsteht:

$$\hat{\theta} = \arg \min_{\theta} \|\theta\|_1, \quad \text{sodass} \quad \mathbf{y} = \mathbf{H}\Psi\theta. \quad (5.10)$$

Sind die Messungen rauschbehaftet, was in der Praxis meistens der Fall ist, lässt sich das obige Problem mit  $0 < \epsilon \ll 1$  abmildern zu:

$$\hat{\theta} = \arg \min_{\theta} \|\theta\|_1, \quad \text{sodass} \quad \|\mathbf{H}\Psi\theta - \mathbf{y}\|_2 \leq \epsilon. \quad (5.11)$$

Aufgrund der Optimierungstheorie kann das Problem (5.11) zudem nicht nur als Kostenfunktion mit Regularisierungsterm formuliert werden, wie es z. B. im LASSO-Verfahren der Fall ist (vgl. Abschnitt 3.2), sondern auch als duales Problem über die Herleitung mithilfe von Lagrange-Multiplikatoren mit einer angepassten Nebenbedingung  $\mathcal{H}(\tilde{\mathbf{x}}) \leq 0$  durch

$$\hat{\theta} = \arg \min_{\theta} \|\mathbf{H}\Psi\theta - \mathbf{y}\|_2, \quad \text{sodass} \quad \|\theta\|_1 \leq \epsilon', \quad (5.12)$$

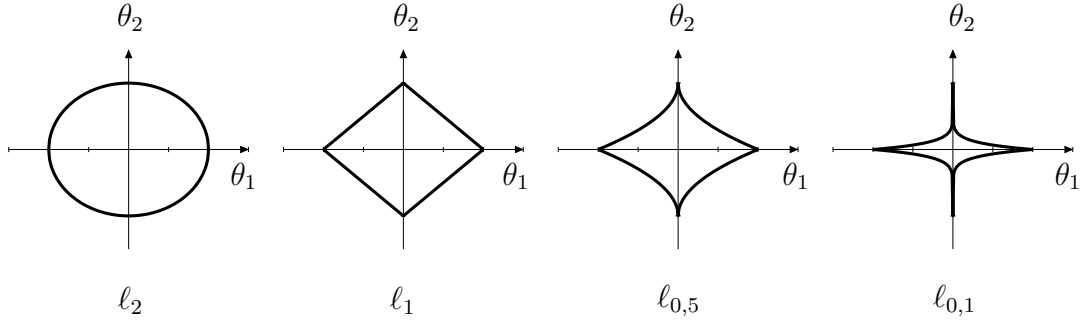
aufgefasst werden [201], [202]. Innerhalb einer iterativen, auf der Lösung einer  $\ell_2$ -Kostenfunktion basierenden Filterstruktur stört jedoch eine Nebenbedingung  $\mathcal{H}$ . Diese kann daher als eine Projektion auf den Lösungsraum interpretiert werden [201], [202]. Dazu wird die Ungleichheitsbedingung durch eine zusätzliche, fiktive Messung mit  $\epsilon' > 0$  und

$$0 = \|\theta\|_1 - \epsilon' \quad (5.13)$$

innerhalb des Filters umgesetzt, die schließlich durch die vorhandenen Strukturen im Filter ausgewertet werden kann. Der Vorteil des  $\ell_1$ -basierten Optimierungsproblems in den Gleichungen (5.10) und (5.11) sowie des  $\ell_1$ -restringierten Problems in Gleichung (5.12) besteht in der Garantie der Konvexität, während gleichzeitig eine dünnbesetzte Lösung gefördert wird. In der Abbildung 5.3 wird der Zusammenhang zwischen den Eigenschaften Konvexität und Sparsity illustriert, indem für

<sup>7</sup>Die Gleichung (5.9) ist ein *nicht-polynomiales, hartes (NP-hard)* Problem [14], [28].

einen zweidimensionalen Parameter die durch die jeweilige  $\ell_p$ -Norm induzierte Region dargestellt ist. Je kleiner das  $p$  der Norm gewählt ist, desto mehr werden das Konzept der Sparsity berücksichtigt und dünnbesetzte Parameter gefördert. Allerdings ist die induzierte Region lediglich für die  $\ell_2$ - und  $\ell_1$ -Norm konvex, während dies für die Regionen mit  $p < 1$  nicht mehr gilt. Die  $\ell_1$ -Norm stellt demnach den besten Kompromiss zwischen Konvexität und Sparsity dar. Die Umstände, welche eine äquivalente Rekonstruktion des Signals durch die Nutzung der  $\ell_1$ -Norm erlauben, sind daher geometrischer Natur und sollen annähernd einer unitären Transformation entsprechen [14], [28], [195], [199], [200].



**Abbildung 5.3** Zusammenhang zwischen der  $\ell_p$ -Norm und den Eigenschaften Konvexität sowie Sparsity: Je kleiner  $p$  ist, desto mehr nähern sich die Parameter der Eigenschaft Sparsity an. Gilt jedoch  $p < 1$ , ist keine Konvexität mehr gegeben (vgl. [14], [17]).

Um ein Optimierungsproblem in der Art von Gleichung (5.12) in der klassischen Korrektor- und Prädiktorstruktur eines SRUKFs zu formulieren, wird das Modell (5.1) zunächst mit einer Schrittweite von  $\Delta t > 0$  diskretisiert und hinsichtlich des SRUKFs angepasst, indem noch Prozess- und Messrauschen in der Formulierung der Dynamiken  $\mathbf{f}$  bzw.  $\tilde{\mathbf{f}}$  berücksichtigt werden:

$$\begin{aligned} \mathbf{x}_{k+1} &= \mathbf{x}_k + \Delta t \cdot (\mathbf{f}(\mathbf{x}_k, \mathbf{u}_k) + \mathbf{w}_k^x) \quad \Rightarrow \quad \tilde{\mathbf{x}}_{k+1} \approx \tilde{\mathbf{x}}_k + \Delta t \cdot \left( \tilde{\mathbf{f}}(\tilde{\mathbf{x}}_k, \mathbf{u}_k) + \begin{bmatrix} \mathbf{w}_k^x \\ \mathbf{w}_k^\theta \end{bmatrix} \right), \\ y_k &= \mathbf{h}(\mathbf{x}_k, \mathbf{u}_k) + \mathbf{v}_k. \end{aligned} \quad (5.14)$$

Dabei gilt für das Prozessrauschen  $\mathbf{w}_k^x \in \mathbb{R}^n \sim \mathcal{N}(\mathbf{0}, \mathbf{Q}_x)$ ,  $\mathbf{w}_k^\theta \in \mathbb{R}^{n_\theta} \sim \mathcal{N}(\mathbf{0}, \mathbf{Q}_\theta)$  und für das Messrauschen  $\mathbf{v}_k \sim \mathcal{N}(\mathbf{0}, \mathbf{R})$ . Somit kann die Prozesskovarianzmatrix für den erweiterten Zustand vereinfachend als Blockmatrix der beiden Einzelkovarianzen durch  $\tilde{\mathbf{Q}} = \text{blkdiag}(\mathbf{Q}_x, \mathbf{Q}_\theta)$  mit den restlichen Einträgen als Null angenommen werden. Zur Übersichtlichkeit werden jedoch in den beiden folgenden Gleichungen die Zeitindizes  $k$  vernachlässigt. Ausgehend von der Minimierung des Schätzfehlers sowie den Überlegungen im vorigen Abschnitt, vgl. Gl. (5.12), kann das Schätzproblem mit gleichzeitiger Dünnbesetztheit der Parameter durch

$$\hat{\tilde{\mathbf{x}}}^* = \arg \min_{\hat{\tilde{\mathbf{x}}}} \frac{1}{2} \mathbb{E}[(\hat{\tilde{\mathbf{x}}} - \tilde{\mathbf{x}})^T (\hat{\tilde{\mathbf{x}}} - \tilde{\mathbf{x}})], \quad \text{sodass} \quad \|\tilde{\mathbf{I}} \hat{\tilde{\mathbf{x}}}\|_1 \leq \epsilon, \quad (5.15)$$

definiert werden. Dabei stellt  $0 < \epsilon \ll 1$  eine Schranke bzgl. des Messrauschens dar, während  $\tilde{\mathbf{I}} = \text{blkdiag}(\mathbf{0}_n, \mathbf{I}_{n_\theta})$  eine Blockmatrix aus Nullen und Einsen beschreibt

und somit die Dünnbesetztheit der Parameter  $\boldsymbol{\theta}$  kodiert. Allerdings ist die Einbettung und Lösung des Optimierungsproblems (5.15) innerhalb der iterativen Struktur des Filters wie bereits im vorigen Abschnitt angesprochen nicht einfach. Daher wird auf die zusätzliche, fiktive Messung des vorigen Abschnitts (5.13) zurückgegriffen, welche als Projektion aufgefasst werden kann [198], [201], [202]. Diese wird als *Pseudomessung*  $y_{pm}$  mit der angepassten, stetigen Ausgangsgleichung  $h_{pm}$  durch

$$y_{pm} = h_{pm}(\tilde{\mathbf{x}}) = \max(\|\tilde{\mathbf{I}}\tilde{\mathbf{x}}\|_1 - \epsilon, 0) \quad (5.16)$$

definiert. Die Größe  $\epsilon \sim \mathcal{N}(0, R_{pm})$  repräsentiert dabei das fiktionale Messrauschen, welches durch das Optimierungsproblem die Nebenbedingung steuert. Hierbei stellt  $R_{pm}$  die Kovarianz des fiktionalen Messrauschens dar. Da das SRUKF auf der Unscented Transformation basiert (vgl. Unterabschnitt 3.5.2), kann die nichtlineare, fiktionale Ausgangsgleichung (5.16) direkt innerhalb des Filters eingesetzt werden, ohne dass weitere Anpassungen erforderlich sind. Dies steht im Kontrast zum EKF, vgl. [187], [202]. Da es sich um eine zusätzliche Messung handelt, ist keine erneute Auswertung des Dynamikschritts erforderlich, sodass stattdessen die Identitätsabbildung  $\mathbf{f}_{Id}$  für diesen Schritt im SRUKF genutzt wird. Bei Bedarf kann die Projektion mittels der Pseudomessung mehrfach erfolgen, um die Genauigkeit der Nebenbedingung  $\mathcal{H}$  zu erhöhen [201], [202]. Ist dies gewünscht, muss die maximale Iteration  $N_{iter} \neq 1$  gewählt werden. Das entwickelte Vorgehen des Joint Estimation SRUKFs (JE-SRUKFs) ist im Algorithmus 2 zusammengefasst und wird nachfolgend kurz erläutert (vgl. [17], [198]).

Aufgrund der harten Grenze  $\tilde{\lambda}$ , welche beschreibt, ab wann ein Element  $\hat{\theta}_i$  als Nichtnullelement bewertet wird, resultiert erfolgreich der dünnbesetzte Parametervektor  $\hat{\boldsymbol{\theta}}$ . Diese Grenze wird ähnlich zum Parameter im LASSO-Verfahren je nach Anwendungsziel bzw. Skalierung der Parameter festgelegt und befindet sich nahe Null. Um jedoch harte Sprünge in den Werten von einem zum nächsten Zeitschritt zu vermeiden, besteht die Option, mittels des Parameters  $\gamma \in [0, 1]$  eine Gewichtung aus dem vorherigen und neuen Wert des Parametervektors  $\hat{\boldsymbol{\theta}}_k$  bzw.  $\hat{\boldsymbol{\theta}}_{pm,j}$  vorzunehmen (vgl. Zeile 10). Im Algorithmus wird dies durch die Indizes  $(n + 1) : \tilde{n}$  dargestellt, wobei  $\tilde{n}$  als Gesamtsystemordnung durch  $\tilde{n} := n + n_{\boldsymbol{\theta}}$  definiert ist. Daraufhin setzt sich der finale, augmentierte Zustand  $\hat{\mathbf{x}}_{k,final}$  durch den zuvor im klassischen Vorgehen berechneten Zustand  $\hat{\mathbf{x}}_k$  und den ggf. gewichteten Parametervektor  $\hat{\boldsymbol{\theta}}_k$  zusammen (vgl. Zeilen 9 und 10). Anschließend wird zunächst wiederum der klassische Algorithmus des SRUKFs durchlaufen, der in jedem Zeitschritt  $k$  einmalig erfolgt. Dies wird durch die letzten Rechenoperationen in den Zeilen 1 und 2 angedeutet, bevor der Pseudocode des Teils folgt, der das Konzept der Sparsity umsetzt (vgl. [17], [198]).

**Dünnbesetztheit mittels stochastischer Modellierung** Eine Alternative zur Modellierung der Dünnbesetztheit besteht in der Ausnutzung der Filterstruktur. Das SRUKF ist stochastisch motiviert und nimmt an, dass die Zufallsvariablen, somit auch der erweiterte Zustand  $\tilde{\mathbf{x}}$ , Gauß-verteilt sind [184], [192]. Das Vorwissen, dass die Parameter der Linearkombination dünnbesetzt sein sollen, lässt sich allerdings nicht nur deterministisch, sondern auch probabilistisch ausdrücken. Mithilfe einer geeigneten Wahrscheinlichkeitsverteilung kann diese Eigenschaft direkt als A-Priori-

---

**Algorithmus 2** JE-SRUKF mit Schranke zur Umsetzung der Dünnbesetztheit

---

```

1:   ⋮
1:    $\hat{\mathbf{x}}_k = \hat{\mathbf{x}}_k^- + \mathbf{K}_k (\mathbf{y}_k - \hat{\mathbf{y}}_k^-)$ 
2:    $\mathbf{S}_k = \text{cholupdate}(\mathbf{S}_k^-, \mathbf{U}, -1)$ 

   % Sparsity-Überprüfung und ggf. Aktualisierung:
3:   Initialisiere:  $h_{pm}, N_{iter}, n_{\theta,act}, \gamma, j = 1, \mathbf{S}_{pm,0} = \mathbf{S}_k, \hat{\mathbf{x}}_{pm,0} = \hat{\mathbf{x}}_k$ 
4:   while  $\#\{\theta_j | \theta_j > \tilde{\lambda}\} > n_{\theta,act}$  and  $j < N_{iter}$ 

       % Schätzung mit Sparsity-Bedingung
5:        $\hat{\mathbf{x}}_{pm,j}, \mathbf{S}_{pm,j} \leftarrow \text{SRUKF mit } (\hat{\mathbf{x}}_{pm,j-1}, \mathbf{S}_{pm,j-1}, \mathbf{f}_{Id}, h_{pm}, \tilde{\mathbf{Q}}, R_{pm})$ 
6:        $j = j + 1$ 
7:   end

       % Bestimmung des finalen Zustands und dessen Kovarianz
8:    $\mathbf{S}_{k,final} = \mathbf{S}_{pm,j}$ 
9:    $[\hat{\mathbf{x}}_{k,final}]_{(1:n)} = [\hat{\mathbf{x}}_k]_{(1:n)},$ 
10:   $[\hat{\mathbf{x}}_{k,final}]_{(n+1:\tilde{n})} = (1 - \gamma)[\hat{\mathbf{x}}_{pm,j}]_{(n+1:\tilde{n})} + \gamma[\hat{\mathbf{x}}_k]_{(n+1:\tilde{n})}$ 
11: end

```

---

Verteilung dem Filter übergeben werden. Als Beispiel gilt die Laplace-Verteilung, welche sich für einen Parameter  $\theta_i$  durch

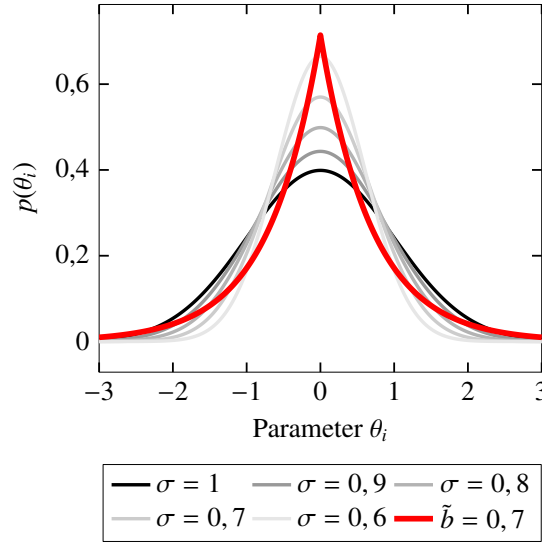
$$p(\theta_i | \mu, \tilde{b}) = \frac{1}{2\tilde{b}} e^{-\frac{|\theta_i - \mu|}{\tilde{b}}} \quad (5.17)$$

mit  $\mu \in \mathbb{R}, \tilde{b} \in \mathbb{R}^+$  charakterisiert [14]. Allerdings sprechen zwei Nachteile gegen die explizite Verwendung dieser Verteilung, da einerseits Gauß-Verteilungen im SRUKF zugrunde gelegt werden und andererseits die A-Posteriori-Verteilung bei der gewählten A-Priori-Verteilung nicht die typische Form der Laplace-Verteilung konserviert [14], [203]. Diese Herausforderung wird in der Dissertation [17] ausführlicher thematisiert. Daher wird eine Gauß-Verteilung zur Modellierung der Parameter genutzt, welche die Eigenschaften der Laplace-Verteilung imitiert. Die Idee dieses Vorgehens ist in der Abbildung 5.4 illustriert. Eine beliebte Wahl ist die Regularized-Horseshoe-Verteilung (RHS-Verteilung), deren Gauß-Verteilung aus unterlagerten Distributionen besteht und folgendermaßen definiert ist [203], [204], [205]:

$$\begin{aligned}
\theta_i | \check{\lambda}_i, \tau, c &\sim \mathcal{N}(0, \check{\lambda}_i^2 \tau^2), \\
\lambda_i &\sim \mathcal{C}^+(0, 1), \\
\tau &\sim \mathcal{C}^+(0, \tau_0), \\
c^2 &\sim \text{Inv}\Gamma(a, b), \\
\check{\lambda}_i &= \frac{c\lambda_i}{\sqrt{c^2 + \tau^2 \lambda_i^2}}.
\end{aligned} \quad (5.18)$$



Hierbei stellen die Parameter  $\tau_0, a, b$  Einflussmöglichkeiten dar, um den Grad der Dünnbesetztheit und der Regularisierung von Maximalwerten einzustellen [204]. In der Regel wird beispielsweise  $\tau_0 \ll 1$  gewählt, da dieser Parameter auf die Dünnbesetztheit der gesamten Verteilung wirkt, während  $\lambda_i$  einige Ausnahmen von dieser Dünnbesetztheit erlaubt. Die inverse Gamma-Verteilung  $\text{Inv}\Gamma$  dient vor allem der Regularisierung von Maximalwerten. Den Einfluss der einzelnen, unterlagerten Distributionen auf die RHS-Verteilung in Abhängigkeit ihrer Parametrierung zeigen Abbildungen in der Dissertation [17].



**Abbildung 5.4** Laplace-Verteilung (visualisiert in rot) im Vergleich zu verschiedenen Gauß-Verteilungen (dargestellt durch Graustufen): Für alle abgebildeten Verteilungen gilt  $\mu = 0$ , vgl. [17], [197].

Wird die stochastische Modellierung des Parametervektors angestrebt, kann analog zum vorherigen Abschnitt weiterhin das augmentierte Modell (5.14) innerhalb des SRUKFs genutzt werden. Zudem bestehen dieselben Annahmen und Definitionen bzgl. der Kovarianzen und der Pseudomessung (5.16). Ebenso findet wie im Algorithmus 2 zunächst ein Durchlauf des Standard-SRUKFs statt. Allerdings unterscheidet sich der Beobachterentwurf nun in der Art und Weise, wie die Dünnbesetztheit für die Parameter umgesetzt wird. Durch die stochastische Modellierung der Parameter  $\theta$  wird in jedem Zeitschritt eine Varianz  $\sigma_\star^2$  mithilfe der RHS-Verteilung bestimmt, um die Laplace-Verteilung zu imitieren. Anschließend werden die Gewichte der UT angepasst und ein erneuter Durchlauf des SRUKFs mit der Identität  $\mathbf{f}_{Id}$  als Dynamikvorschrift und mit der Pseudomessung  $h_{pm}$  als Messmodell vollzogen. Die Anpassung der Gewichte ist erforderlich, da im SRUKF eine Standardnormalverteilung angenommen ist, sodass  $\kappa^{(1)} = 3 - \tilde{n}$  optimal gewählt wird [194], [196]. Ist dies wie bei der stochastischen Modellierung der Parameter  $\theta$  nicht der Fall, gilt für diese Situation  $\kappa^{(2)} = 3\sigma^4 - \tilde{n}$ , wobei  $\sigma$  zu  $\sigma_\star$  aufgrund der RHS-Verteilung bestimmt wird [197]. Anschließend wird eine erneute Schätzung durchgeführt, woraufhin sich der Zustand  $\tilde{\mathbf{x}}_{k+1|k}^-$  aus den  $n$  Einträgen der ersten Schätzung und aus den  $n_\theta$  Einträgen der zweiten Schätzung ergibt (vgl. Zeilen 9 bis 10). Gleiches gilt für die Kovarianz in den Zeilen 7 und 8. Dieses wird jeweils durch die eckigen Klammern und mithilfe der programmiertechnischen Darstellung des Doppelpunkts dargestellt.

Daher resultiert der folgende Algorithmus, welcher sich in jedem Zeitschritt  $k$  dem Durchlauf des Algorithmus eines klassischen SRUKFs anschließt:

---

**Algorithmus 3** JE-SRUKF mit stochastischer Umsetzung der Sparsity
 

---

```

1:  ⋮
    $\hat{\mathbf{x}}_k = \hat{\mathbf{x}}_k^- + \mathbf{K}_k (\mathbf{y}_k - \hat{\mathbf{y}}_k^-)$ 
2:   $\mathbf{S}_k = \text{cholupdate}(\mathbf{S}_k^-, \mathbf{U}, -1)$ 

3:  Initialisiere:  $R_{pm}, \tau_0, a, b$ 

   % Bestimmung der Varianz bzw. der neuen Gewichte
4:   $\sigma_\star^2 = \mathbb{E}[\sigma^2] \leftarrow$  Bestimme mit Gl. (5.18) mit  $\tau_0, a, b$ 
5:   $\mathbf{W}_m^{(2)}, \mathbf{W}_c^{(2)} \leftarrow \alpha, \beta, \kappa^{(2)} = 3\sigma_\star^4 - \tilde{n}$ 

   % Schätzung mit Sparsity-Modellierung
6:   $\hat{\mathbf{x}}_{pm}, \mathbf{S}_{pm} \leftarrow$  SRUKF mit  $(\hat{\mathbf{x}}_k, \mathbf{S}_k, \mathbf{f}_{Id}, h_{pm}, \tilde{\mathbf{Q}}, R_{pm})$ 

   % Bestimmung des finalen Zustands und dessen Kovarianz
7:   $\mathbf{S}_{k,final} = \mathbf{S}_k$ 
8:   $[\mathbf{S}_{k,final}]_{(n+1:\tilde{n}, n+1:\tilde{n})} = \mathbf{S}_{pm}$ 

9:   $\hat{\mathbf{x}}_{k,final} = \hat{\mathbf{x}}_k,$ 
10:  $[\hat{\mathbf{x}}_{k,final}]_{(n+1:\tilde{n})} = [\hat{\mathbf{x}}_{pm}]_{(n+1:\tilde{n})}$ 

11: end

```

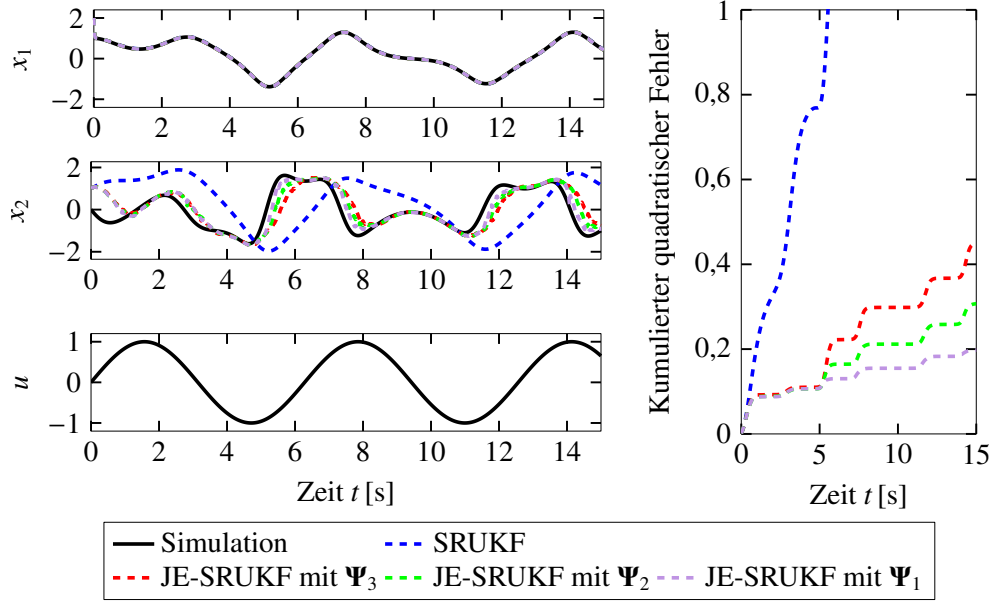
---

### 5.2.2 Beispielhafte Anwendungen

Das zuvor erläuterte Konzept des augmentierten Beobachters inklusive seiner zwei Möglichkeiten, die Dünnbesetztheit der Parameter  $\boldsymbol{\theta}$  umzusetzen, wird nachfolgend durch zwei Anwendungsbeispiele illustriert. Dabei wird die Schätzgüte sowie der Einfluss der Bibliothek  $\Psi$  auf das Schätzverhalten deutlich. Die dargestellten Beispiele dienen allerdings nur zur Veranschaulichung, sodass für weitere Details und Anwendungen auf die Dissertation [17] verwiesen wird.

**Duffing-Oszillator** Um die Wirkungsweise des augmentierten Beobachters zu veranschaulichen, wird nachfolgend der Duffing-Oszillator eingeführt, welcher üblicherweise als Stellvertreter vieler dynamischer Systeme untersucht wird [17], [28]. Es handelt sich um ein System zweiter Ordnung, welches beispielsweise den Einfluss einer Kraft auf einen Körper beschreibt:

$$\begin{aligned} \dot{\mathbf{x}} &= \begin{bmatrix} x_2 \\ -\theta_3 x_2 - \theta_1 x_1 - \theta_2 x_1^3 + u \end{bmatrix}, \\ y &= x_1. \end{aligned} \tag{5.19}$$



**Abbildung 5.5** Qualität der Zustandsschätzung im Vergleich zu verschiedenen Bibliotheken und einem klassischen SRUKF, das ohne die Modellungenauigkeit  $g$  schätzt, wenn die Umsetzung der Dünnbesetztheit mittels einer Schranke gewählt worden ist, vgl. [17], [198]

Hierbei können die Zustände  $\mathbf{x} = [x_1, x_2]^T \in \mathbb{R}^2$  exemplarisch Winkel und Winkelgeschwindigkeit sowie der Eingang  $u \in \mathbb{R}$  eine Kraft darstellen. Die physikalischen Parameter werden für alle folgenden Untersuchungen zu  $\boldsymbol{\theta} = [1, -3, 0, 1]^T$  gewählt. Ferner werden der kubische Term als Modellungenauigkeit  $g(\mathbf{x}, u) = -\theta_2 x_1^3 = -3x_1^3$  angenommen und für die Identifikation dieser Ungenauigkeit drei sich nur leicht unterscheidende Bibliotheken festgelegt:

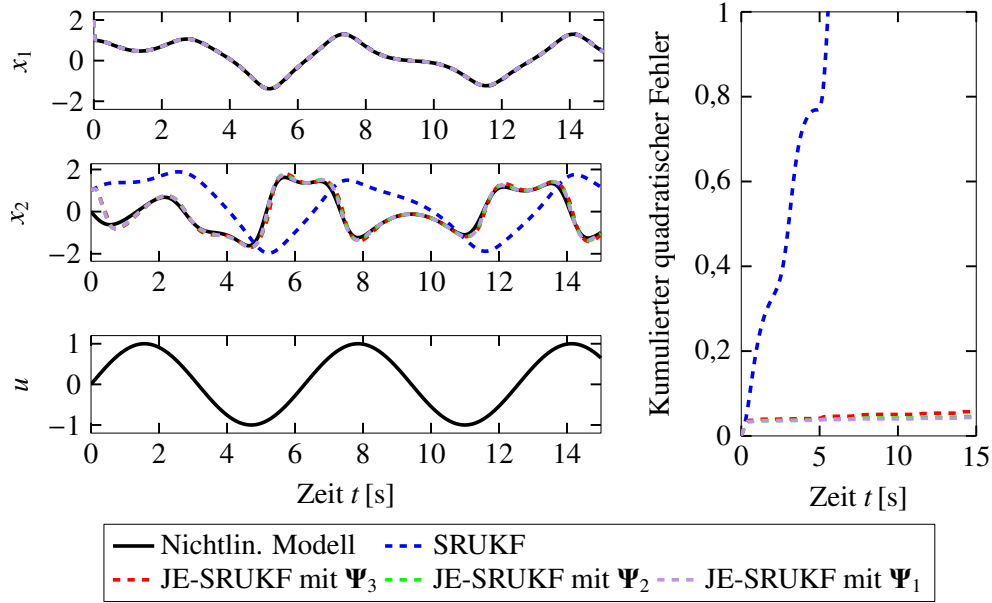
$$\Psi_1(\mathbf{x}, u) = [1, x_1, x_2, x_2^2, \sin(x_2), x_1 \cdot x_2, \cos(x_1), u, x_1^3]^T, \quad (5.20)$$

$$\Psi_2(\mathbf{x}, u) = [1, x_1, x_2, x_2^2, \sin(x_2), x_1 \cdot x_2, \cos(x_1), u, x_1^2]^T, \quad (5.21)$$

$$\Psi_3(\mathbf{x}, u) = [1, x_1, x_2, x_2^2, \sin(x_2), x_1 \cdot x_2, \cos(x_1), u]^T. \quad (5.22)$$

Hierbei weist lediglich die erste Bibliothek den gesuchten kubischen Term auf, während die anderen beiden jeweils den quadratischen bzw. linearen Term als Funktionskandidaten ansetzen. In den Abbildungen 5.5 und 5.6 sind die geschätzten Zustände im Vergleich zu den simulierten Zuständen bei einer sinusförmigen Anregung in Abhängigkeit der verschiedenen Bibliotheken dargestellt. Das Szenario, welches in den beiden Abbildungen zu sehen ist, unterscheidet sich lediglich durch die gewählte Strategie zur Umsetzung der Dünnbesetztheit der Parameter. In beiden Abbildungen ist zu erkennen, dass ein klassisches SRUKF (blau) ohne Kenntnis der Modellungenauigkeit nicht in der Lage ist, korrekte Schätzwerte zu liefern. Im Gegensatz dazu ermöglicht der augmentierte Beobachter in beiden Fällen eine hohe Schätzgüte durch die Approximation mittels Linearkombinationen. Die Schätzgüte nimmt dabei geringfügig je nach gewählter Bibliothek und/oder Umsetzung der Dünnbesetztheit ab, ist aber besonders hoch, wenn die Bibliothek den tatsächlichen, kubischen Term aufweist und die stochastische Strategie zur Dünnbesetztheit gewählt wird.

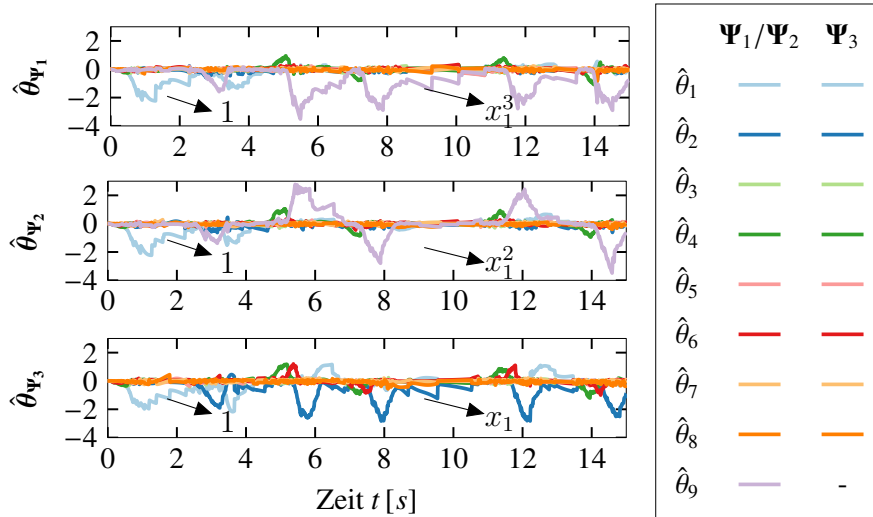
Neben dem primären Ziel, eine hohe Schätzgüte zu erzielen, ist es ebenso auf-



**Abbildung 5.6** Qualität der Zustandsschätzung im Vergleich zu verschiedenen Bibliotheken und einem klassischen SRUKF, das ohne die Modellungenauigkeit  $g$  schätzt, wenn die stochastische Umsetzung der Dünnbesetztheit gewählt worden ist, vgl. [17], [197]

schlussreich, woher die Modellungenauigkeit resultiert. Um einen Einblick in den Charakter der Modellungenauigkeit zu gewinnen, kann der Verlauf der Parameter-elemente  $\theta_i$  mit  $i = 1, \dots, n_\theta$  analysiert werden (vgl. dazu auch Abschnitt 5.3). Für beide Umsetzungen ist dies jeweils in den Abbildungen 5.7 und 5.8 durchgeführt worden. Auffällig ist hierbei auf den ersten Blick, dass beide Umsetzungen ähnliche Erkenntnisse liefern, aber die Verläufe der stochastischen Umsetzung wesentlich glatter sind, da die Dünnbesetztheit in diesem Fall global innerhalb der Filterstruktur angestrebt wird. Beide Strategien führen jedoch auf dieselben Terme, die als diejenigen Bibliotheksfunktionen identifiziert werden, welche die Modellungenauigkeit am besten charakterisieren. Dabei findet der augmentierte Beobachter erwartungsgemäß den kubischen Term, sofern dieser in der Bibliothek enthalten ist. Bemerkenswerterweise werden jedoch auch Alternativen gefunden, falls dieser Term nicht verfügbar ist. Dabei wird insbesondere bei Nutzung des quadratischen Terms das Vorzeichen des kubischen Terms durch einen ständigen Vorzeichenwechsel des Parameters kompensiert (vgl. zweite Zeile der Abbildungen). Da in den dargestellten Algorithmen 2 und 3 nur die jeweilige letzte Messung als Information genutzt wird, ist es zudem empfehlenswert, im Sinne der rekursiven Least-Squares-Methode mehrere zurück liegenden Messungen zu verwenden, um eine eindeutige Konvergenz zu einem Parameterwert zu erhalten.

Da es sich bei dem Duffing-Oszillator um ein bekanntes System handelt und die Modellungenauigkeit  $g$  durch den kubischen Term nur zur Visualisierung angenommen worden ist, kann anschließend die Qualität der Approximation überprüft werden. Qualitativ ist dies beispielsweise durch den zeitlichen Vergleich der Modellungenauigkeit  $g$  und der jeweiligen Approximation  $\hat{g}_i$  mit  $i = 1, 2, 3$  möglich. In der Abbildung 5.9 ist dieser Vergleich anhand der unterschiedlichen Umsetzungsstrategien dargestellt. Die Approximationen erfassen allesamt qualitativ den zu identifizie-



**Abbildung 5.7** Zeitlicher Verlauf des Parametervektors  $\hat{\theta}$  im Vergleich verschiedener Bibliotheken, wenn die Umsetzung der Dünnspeichertechnik mittels einer Schranke gewählt worden ist, vgl. [17], [198], und die jeweils daraus zugeordneten dominanten Bibliotheksterme (dargestellt anhand der Pfeile)

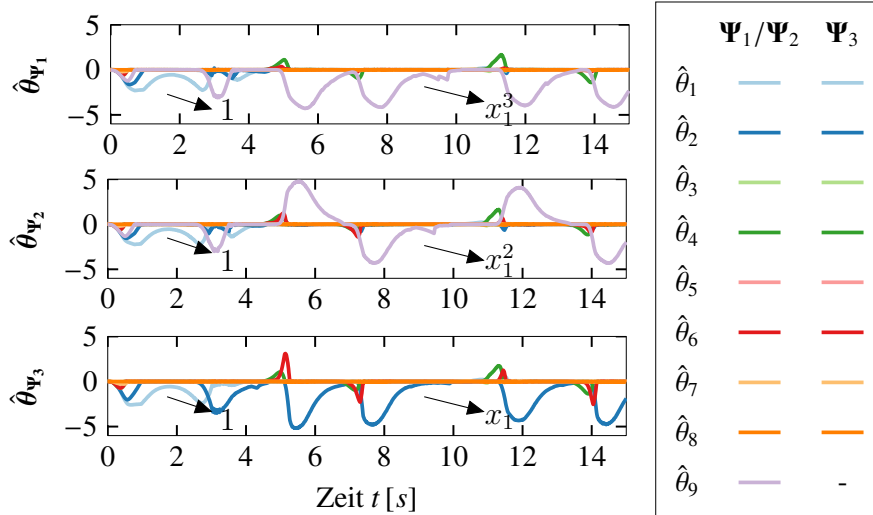
renden charakteristischen Verlauf der Modellungenauigkeit, wobei sich weiterhin die zuvor beobachteten Phänomene, die die Modellgüte beeinflussen, in Abhängigkeit der Bibliotheken und der Umsetzungsstrategie zeigen.

**Golfroboter** Der Golfroboter, welcher im Abschnitt 6.1 vorgestellt wird, kennzeichnet sich durch eine nichtlineare Reibung, den sogenannten Stick-Slip-Effekt. Da dieser aufwendig zu modellieren ist, wird im Folgenden angenommen, dass die Reibung vollständig durch den augmentierten Beobachter geschätzt werden soll. Als Zustandsbeobachter in der Abbildung 6.7 wird daher statt des Luenberger-Beobachters nun der augmentierte Beobachter mit fester Schranke gewählt. Für die Bibliothek empfiehlt sich auf Erfahrungswissen zurückzugreifen und etwaige Hypothesen bzgl. einer Approximation der Reibung als Kandidatenfunktionen hinzuzufügen. Daher kann beispielsweise die folgende Bibliothek genutzt werden, welche eine Mischung aus Polynomen und trigonometrischen Termen aufweist, um die Haft- und Gleitreibung zu erfassen:

$$\Psi(\mathbf{x}, u) = [1, x_1, x_2, x_1^2, x_2^2, x_1 x_2, \cos(x_1), \cos(x_2), \tanh(x_1), \tanh(x_2), u]^T. \quad (5.23)$$

Der Vorteil des augmentierten Beobachters zeigt sich insbesondere in Situationen der Systemveränderungen. Ist die Masse des Golfschlägers exemplarisch verdoppelt worden, ohne dass das Modell des Golfroboter angepasst wurde (vgl. [70]), so zeigt sich die Auswirkung dieser Systemveränderung auf die Schätz- und Regelgüte in der Abbildung 5.10.

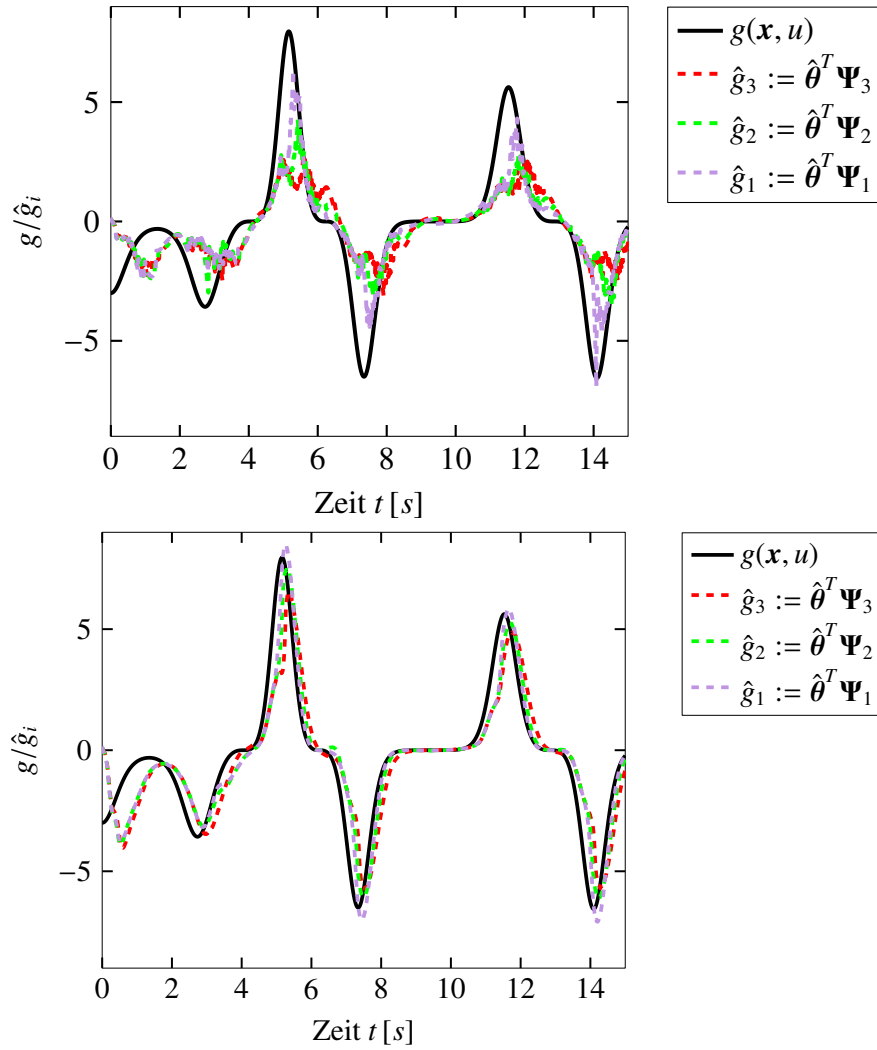
Da die Vorsteuerung und der Regler weiterhin mit dem unveränderten, nichtlinearen Modell (6.11) berechnet werden, ist aufgrund der Systemveränderung eine Abweichung von den Solltrajektorien und ein verändertes Motormoment zu erkennen.



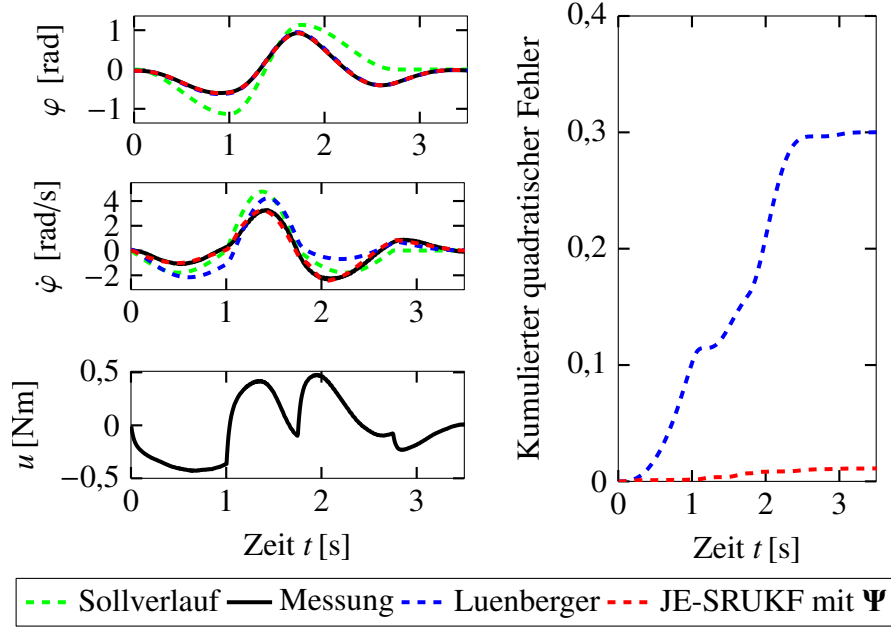
**Abbildung 5.8** Zeitlicher Verlauf des Parametervektors  $\hat{\theta}$  im Vergleich verschiedener Bibliotheken, wenn die stochastische Umsetzung der Dünnbesetztheit gewählt worden ist, vgl. [17], [197], und die jeweils daraus zugeordneten dominanten Bibliotheksterme (dargestellt anhand der Pfeile)

Allerdings gelingt es dem JE-SRUKF mit der Bibliothek  $\Psi$ , die Ist-Trajektorien sehr genau zu schätzen. Im Gegensatz dazu ist es dem Luenberger-Beobachter aufgrund seiner nicht-adaptiven Struktur nicht möglich, die Geschwindigkeit des Schlags korrekt zu erfassen, was der kumulierte quadratische Fehler in der rechten Visualisierung bestätigt. Stattdessen müssten die linearisierten Modelle des Luenberger-Beobachters rekursiv angepasst werden, um ebenso geeignet auf Systemveränderungen reagieren zu können (vgl. [1], [70]). Somit zeigt dieses Experiment den Vorteil der Nutzung des JE-SRUKFs auf: Potentielle Systemveränderungen stellen bei geeignet gewählter Bibliothek keine Herausforderung dar, sodass eine hohe Schätzgüte gewährleistet werden kann. Abschnitt 5.3 zeigt ferner auf, wie eine Modelladaption basierend auf den Daten des augmentierten Beobachters durchgeführt werden kann, welche im Fall einer Systemveränderung zur Robustheit des Beobachters beiträgt und für Vorsteuerung und Regler vorteilhaft sein kann. Die Anwendung am Golfroboter belegt daher zuverlässige Schätzungen in der Prüfstands- und Echtzeitanwendung im geschlossenen Regelkreis. Allerdings ist bei jedem Anwendungsbeispiel der Einfluss der Bibliothek entscheidend für die Stabilität und Schätzgüte des Beobachters, da bei einer zu einseitigen oder klein gewählten Bibliothek ein divergierendes Verhalten auftreten kann (vgl. Untersuchungen in der Dissertation [17]). Folglich sollte die Bibliothek sorgfältig festgelegt werden.

**Windenergieanlage** Die bisherigen Anwendungsbeispiele kennzeichnen sich durch rein additive Modellungenauigkeiten, welche z. B. aufgrund von Dämpfung oder Reibung resultieren. In der Modellierung von Windenergieanlagen spielt jedoch der Einfluss des Windes eine große Rolle, dessen Geschwindigkeit als Störgröße und daher als Modellungenauigkeit modelliert wird. Als Exempel dient ein zweidimensionales Modell einer Windenergieanlage aus der Dissertation [207] bzw. der dazugehörigen Publikation [208], auf deren Grundlage das Referenzmodell simuliert und



**Abbildung 5.9** Approximation der Modellungenauigkeit  $g(x, u) = -3x_1^3$  im Vergleich verschiedener Bibliotheken und der unterschiedlichen Umsetzung bzgl. der Dünnbesetztheit, vgl. [17], [197]: Dünnbesetztheit mittels Schranke (links) und Dünnbesetztheit mittels stochastischer Formulierung (rechts)



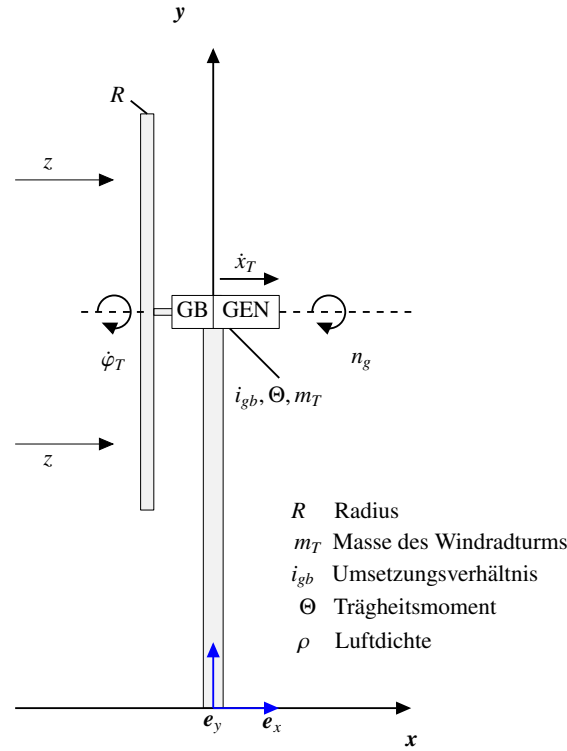
**Abbildung 5.10** Qualität der Zustandsschätzung im Vergleich zum Luenberger-Beobachter im geschlossenen Regelkreis bei Systemveränderungen und Verwendung der Bibliothek  $\Psi$  am Prüfstand, wenn die Umsetzung der Dünnbesetztheit mittels Schranke gewählt worden ist, vgl. [17], [206]

reale Winddaten genutzt werden können. Eine Zeichnung des Systems mit den relevanten physikalischen Größen ist in der Abbildung 5.11 abgebildet. Der Zustand der Windenergieanlage besteht dabei aus dem Rotorwinkel  $\varphi_T$  und der Position der Gondel  $x_T$  sowie deren Geschwindigkeiten, sodass  $\mathbf{x} = [\varphi_T, \dot{\varphi}_T, x_T, \dot{x}_T]^T$  gilt. Daneben wird die Windgeschwindigkeit in  $x$ -Richtung als Störgröße  $z$  modelliert. Somit kann das Modell der zweidimensionalen Windenergieanlage nach [207], [208] abstrahiert und definiert werden durch

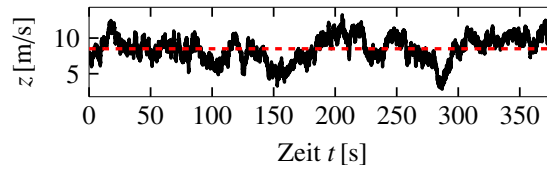
$$\begin{aligned} \dot{\mathbf{x}} &= \begin{bmatrix} x_2 \\ p_1 \cdot C_M(x_2, x_4, z) \cdot (z - x_4)^2 - p_2 u \\ x_4 \\ p_3 \cdot C_T(x_2, x_4, z) \cdot (z - x_4)^2 - p_4 x_4 - p_5 x_3 \end{bmatrix}, \\ \mathbf{y} &= [p_6 x_2, p_3 \cdot C_T(x_2, x_4, z) \cdot (z - x_4)^2 - p_4 x_4 - p_5 x_3] \\ &= [n_g, \dot{x}_4]. \end{aligned} \quad (5.24)$$

Die aerodynamischen Eigenschaften des Rotors werden dabei durch die Koeffizienten der Momente und des Schubs  $C_M(x_2, x_4, z)$  bzw.  $C_T(x_2, x_4, z)$  dargestellt, welche aus einer geschwindigkeits- und drehzahlbasierten Relation resultieren. Kurzdetaill zu diesen Zusammenhängen sowie die Parameterwerte  $p_1, \dots, p_6$  finden sich in der Dissertation [17]. Für ausführliche Informationen wird auf [207], [208] verwiesen. Die Geschwindigkeit des Generators  $n_g$  sowie die Beschleunigung der Gondel  $\ddot{x}_T$  stellen die Messgrößen dar. Als Eingang dient das Moment  $u$  des elektrischen Generators (vgl. [208]). Ziel ist es, die Windgeschwindigkeit, welche in der Abbildung 5.12 dargestellt ist, sowie die Position des Turmkopfes, dessen Geschwindigkeit und die Winkelgeschwindigkeit des Rotorwinkels zu schätzen, um die Windenergieanlage sicher





**Abbildung 5.11** Seitenansicht einer Windenergieanlage, mit freundlicher Genehmigung von B. Ritter [207]



**Abbildung 5.12** Beispielhafter Verlauf von Realdaten der Windgeschwindigkeit  $z$  in  $x$ -Richtung mit Mittelwert in rot und freundlicher Genehmigung von B. Ritter [207].

betreiben zu können. Obwohl die Windenergieanlage ein eingangsaffines System ist und in die Form (5.4) transformiert werden kann, stellt es aufgrund der multiplikativen Ungenauigkeit, die durch den Term  $(z - x_4)^2$  entsteht, ein komplexes und herausforderndes Exempel im Vergleich zu den vorigen Applikationen dar. Um diese Komplexität zu reduzieren, ist es im Beobachterentwurf üblich, ein Störmodell für eine Störung anzunehmen [24]. Dieses zusätzliche Vorwissen kann anschließend in das augmentierte Modell (5.4) eingebracht werden. Aufgrund der Struktur der Winddaten, welche in der Abbildung 5.12 abgebildet sind und um einen Mittelwert von etwa 8,5 m/s variieren, ist es ratsam, ein konstantes Störmodell anzunehmen, sodass  $\dot{z} = 0$  gilt. Wird die Windgeschwindigkeit wiederum als Linearkombination aus geeigneten Bibliotheksfunktionen angenähert (vgl. Gleichung (5.2)), so ergibt sich folgender Zusammenhang, wobei aufgrund der Übersichtlichkeit die Abhängigkeiten der Bibliothek  $\Psi(\mathbf{x}, u)$  vernachlässigt werden:

$$\dot{z} = 0 \quad (5.25)$$

$$\Leftrightarrow \frac{\partial}{\partial t} (\boldsymbol{\theta}^T \boldsymbol{\Psi}) = 0 \quad (5.26)$$

$$\Leftrightarrow \dot{\boldsymbol{\theta}}^T \boldsymbol{\Psi} + \boldsymbol{\theta}^T \dot{\boldsymbol{\Psi}} = 0 \quad (5.27)$$

$$\Leftrightarrow \dot{\boldsymbol{\theta}}^T \boldsymbol{\Psi} = -\boldsymbol{\theta}^T \dot{\boldsymbol{\Psi}} \quad (5.28)$$

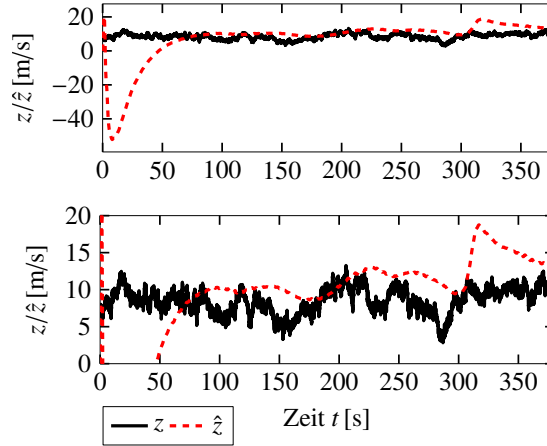
$$\Leftrightarrow \boldsymbol{\Psi}^T \dot{\boldsymbol{\theta}} = -\dot{\boldsymbol{\Psi}}^T \boldsymbol{\theta} \quad (5.29)$$

$$\Leftrightarrow (\boldsymbol{\Psi}^T)^+ \boldsymbol{\Psi}^T \dot{\boldsymbol{\theta}} = -(\boldsymbol{\Psi}^T)^+ \dot{\boldsymbol{\Psi}}^T \boldsymbol{\theta} \quad (5.30)$$

$$\Leftrightarrow \dot{\boldsymbol{\theta}} = -(\boldsymbol{\Psi}^T)^+ \dot{\boldsymbol{\Psi}}^T \boldsymbol{\theta}. \quad (5.31)$$

Anschließend wird die Bibliothek  $\boldsymbol{\Psi}(\mathbf{x}, u, t) = [1, \cos(20 \cdot t), 0,001 \cdot t, x_1, x_3]^T$  gewählt, welche neben den üblichen Konstanten und Zuständen auch Zusammenhänge zur Zeit aufweist, da z. B. Korrelationen zwischen der Biegung des Turms, entsprechend  $x_3$ , und der Windgeschwindigkeit zu vermuten sind. Hierbei ist anzumerken, dass diese Bibliothek entgegen der Formulierung (5.5) zur Vereinfachung und Übersichtlichkeit nur einen Teil der Zustände enthält, da die Bibliothek  $\boldsymbol{\Psi}$  aufgrund der Berücksichtigung des zusätzlichen Vorwissen nach Gleichung (5.25) differenziert werden muss. Kommen die Zustände  $x_2$  und  $x_4$  hinzu, ergibt sich ein sehr ähnliches Resultat. In der Dissertation [17] wird die Schätzgüte anhand der Zustandsverläufe analysiert, welche ausreichend genau trotz der Störung  $z$  ist, jedoch aufgrund der multiplikativen Struktur Beeinträchtigungen aufweist. Erkennbar sind diese Limitationen ebenfalls an der Approximationsgenauigkeit der Windgeschwindigkeit  $z$ . Wird die Linearkombination, welche  $\hat{z}$  nach Glättung des Signals darstellt, nun mit der Windgeschwindigkeit  $z$  verglichen, zeigt die obere Grafik in Abbildung 5.13, dass die Approximation qualitativ gelingt, da sich nach etwa 50 Sekunden eine Konvergenz zum Mittelwert der Windgeschwindigkeit  $z$  einstellt. Allerdings zeigt die untere Grafik in der Vergrößerung, dass der qualitative Verlauf mit einzelnen Schwingungen gut abgebildet wird, aber phasenverzögert erfolgt. Dies erlaubt daher keine exakte Approximation der Windgeschwindigkeit. Das Verfahren des augmentierten Beobachters weist folglich Grenzen bzgl. multiplikativer, rauschbehafteter Modellungenauigkeiten auf, da keine Verbesserung im Vergleich zu bestehenden Verfahren erreicht werden konnte [207], [208]. Dennoch zeigt der Ansatz des JE-SRUKFs für eine solche Ungenauigkeit das vielversprechende Potential für weitere Anwendungen auf, sofern die Formulierung des Beobachters auf allgemeinere Systemklassen erweitert wird.

Der augmentierte Beobachter ist für **eingangsaffine Systeme** anwendbar, welche **additive** Modellungenauigkeiten aufweisen. Es gibt zwei unterschiedliche Umsetzungen, um dünnbesetzte Parameter zu erhalten. Bei **geeignet gewählter** Bibliothek und sichtbarer Ungenauigkeit können eine **hohe Schätzgüte** sowie eine **physikalisch interpretierbare**, parametrische Darstellung der Ungenauigkeit erzielt werden. Besonders vorteilhaft ist der Einsatz bei **Systemveränderungen**, andernfalls ist der Aufwand dem Nutzen im Vergleich zu herkömmlichen Beobachterttypen abzuwägen.



**Abbildung 5.13** Approximation  $\hat{z}$  im Vergleich zur gemessenen Windgeschwindigkeit  $z$  (oben), Vergrößerung (unten), vgl. [17]

### 5.2.3 Augmentierter Sliding-Mode-Beobachter

Im ersten Abschnitt dieses Kapitels wurde bereits der Sliding-Mode-Beobachter vorgestellt, welcher aufgrund seiner Robustheit häufig in der Situation von Modellungenauigkeiten eingesetzt wird. Daher bietet es sich an, die grundlegende Idee der Joint Estimation auf diesen Beobachterttyp zu übertragen. Seine grundlegende Struktur kennzeichnet sich durch

$$\begin{aligned} \begin{bmatrix} \dot{\hat{x}}_1 \\ \dot{\hat{x}}_2 \\ \vdots \\ \dot{\hat{x}}_n \end{bmatrix} &= \begin{bmatrix} \hat{x}_2 + \nu_1(e_y) \\ \hat{x}_3 + \nu_2(e_y) \\ \vdots \\ \hat{f}(\hat{x}_1, \hat{x}_2, \dots, \hat{x}_{n-1}, y) + \nu_n(e_y) \end{bmatrix}, \\ \hat{y} &= \hat{x}_1, \\ e_y &= \hat{y} - y, \end{aligned} \quad (5.32)$$

wobei  $e_y$  den Ausgangsfehler bezeichnet. Mithilfe der Darstellung (5.32) lässt sich die Fehlerdynamik des SMOs ermitteln. Der Modellfehler  $\Delta f$ , der aus Anfangsfehlern, Störungen oder Modellungenauigkeiten resultieren kann, ist dabei als Differenz zwischen Modell  $\hat{f}$  und realem System  $f$  durch

$$\Delta f = \hat{f}(\hat{x}_1, \hat{x}_2, \dots, \hat{x}_{n-1}, y) - f(x_1, x_2, \dots, x_{n-1}, y) \quad (5.33)$$

definiert [188], [189], [190]. Um nicht nur eine dynamische Approximation der Ungenauigkeiten mittels eines Tiefpassfilters, sondern auch eine parametrische, physikalisch nachvollziehbare Identifikation zu erzielen, kann die Grundidee der Augmentation auf diesen Typ Zustandsbeobachter übertragen und um eine automatisch gewählte Bibliothek ergänzt werden. Im Fall eines SMOs kann die Modellungenauigkeit  $\Delta f$  oder Störung durch die Struktur des Beobachters und der daraus resultierenden Schätzfehlerdynamik direkt in Abhängigkeit der  $n$ -ten Schaltfunktion  $\nu_n$  formuliert werden:

$$\Delta f = -\nu_n(e_y). \quad (5.34)$$

Üblicherweise wird diese Ungenauigkeit in Gleichung (5.34) mithilfe eines Tiefpassfilters approximiert. Infolge der vorangegangenen Abschnitte lässt sich diese jedoch auch folgendermaßen definieren:

$$\Delta f = -\nu_n(e_y) = \boldsymbol{\theta}^T \boldsymbol{\Psi}(\mathbf{x}, \mathbf{u}), \quad (5.35)$$

indem erneut eine Linearkombination aus einer geeignet gewählten Bibliothek  $\boldsymbol{\Psi}$  gewählt wird. Um optimale Parameterwerte zu finden, wird der Fehler

$$\mathbf{e}_\theta = -\nu_n(e_y) - \boldsymbol{\theta}^T \boldsymbol{\Psi}(\mathbf{x}, \mathbf{u}) \quad (5.36)$$

minimiert, dessen Lösung durch

$$\hat{\boldsymbol{\theta}} = \left( - \int_0^t \nu_n(e_y) \boldsymbol{\Psi}(\hat{\mathbf{x}}, \mathbf{u})^T d\tau \right) \left[ \int_0^t \boldsymbol{\Psi}(\hat{\mathbf{x}}, \mathbf{u}) \boldsymbol{\Psi}(\hat{\mathbf{x}}, \mathbf{u})^T d\tau \right]^{-1} \quad (5.37)$$

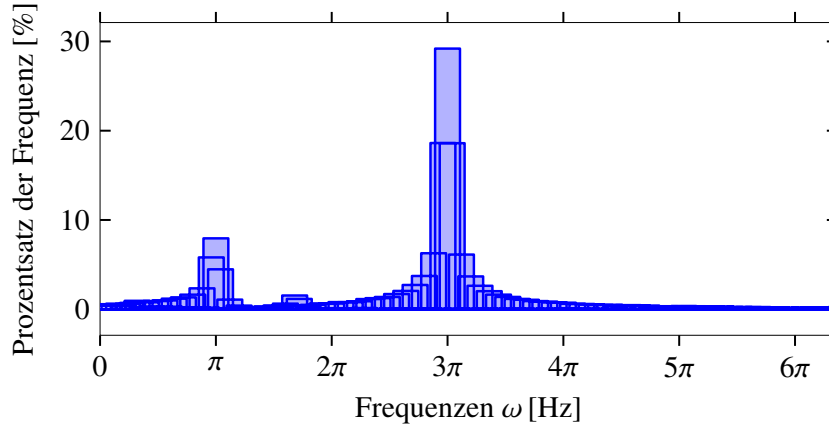
bestimmt werden kann. Diese Lösung wird unter der Annahme, dass der SMO gut parametrisiert ist und konvergiert, über einen rekursiven Least-Squares-Ansatz mithilfe einer dynamischen Berechnung der Inversen ermittelt [188]. Neben der Berücksichtigung von Vorwissen in Form von Hypothesen oder der Nutzung der minimalen Bibliothek (5.5) kann jedoch zunächst eine Datenakquise erfolgen, auf deren Grundlage die Charakteristika der Störung analysiert werden [209]. So kann z. B. bei oszillierenden Störungen eine Fouriertransformation genutzt werden, um auftretende Frequenzen aus Verläufen von  $\nu_n(e_y)$  zu identifizieren. Die Fouriertransformation bietet zudem den Vorteil, dass es sich um eine orthonormale Basistransformation handelt (vgl. Abschnitt 5.2). Dazu werden für einen bestimmten Zeithorizont die Daten der Schaltfunktion aufgenommen, während sich der SMO in der Sliding-Phase befindet. Diese Informationen können anschließend verwendet werden, um mittels der Fouriertransformation Frequenzen des vergangenen Zeitraums in Ansatzfunktionen  $\psi_i$  zu platzieren [209]. Somit enthält die Bibliothek Terme, die höchstwahrscheinlich der Identifikation der Störung bzw. Modelldiskrepanz  $\Delta f$  dienen.

**Einfachpendel auf einem Wagen** Das Einfachpendel auf einem Wagen ist ein klassisches, nichtlineares Anwendungsbeispiel in der Regelungstechnik [21], [24]. Im Folgenden wird allerdings angenommen, dass sich eine Störung  $\rho$  auf den Wageneingang  $u$  auswirkt, welche unbekannter Art ist. Der modifizierte SMO soll diese identifizieren, indem das Hilfsmittel der Fourieranalyse genutzt wird. Dazu werden für die Anregung und die Störung

$$\begin{aligned} u(t) &= \sin\left(\pi \cdot t + \frac{\pi}{2}\right), \\ \rho(t) &= 4 \cdot \sin\left(3\pi \cdot t + \frac{\pi}{2}\right). \end{aligned} \quad (5.38)$$

angenommen. Indem die Messdaten des Einfachpendels genutzt und einer Fourieranalyse zugeführt werden, ergibt sich die Abbildung 5.14, welche das durch die Fouriertransformation detektierte Frequenzspektrum anhand der relativen Häufigkeit in den Daten kennzeichnet. Durch die Abbildung 5.14 ist erkennbar, dass die Fou-

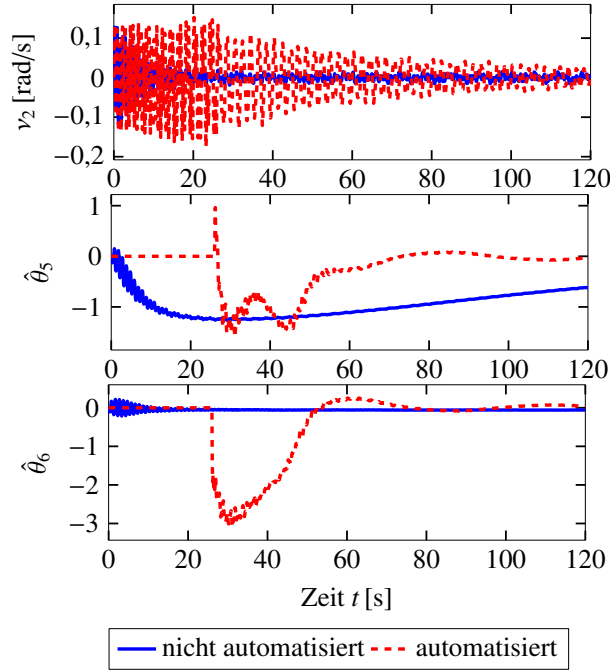
riertransformation die Frequenz der Störung  $\omega_p = 3\pi$  in den Daten als wichtigste Frequenz erkennt, aber darüber hinaus jene der Anregung  $\omega_u = \pi$  als weitere Größe detektiert. Wird dieses Vorgehen mit der Approximation über Gleichung (5.37) gekoppelt, entsteht eine automatische Formulierung der Bibliothek  $\Psi$ , welche zur Schätzung der Ungenauigkeit  $\Delta f$  geschickt beiträgt.



**Abbildung 5.14** Die durch die Fouriertransformation identifizierten Frequenzen zur automatisierten Bildung von Bibliothekstermen  $\psi_i$  weisen die höchsten Prozentsätze auf, vgl. [17], [209].

Exemplarisch ist dieses Verfahren in der Abbildung 5.15 dargestellt, bei dem die Schätzung der Ungenauigkeit mit einer nicht automatisch bestimmten Bibliothek verglichen wird. Es lässt sich anhand des Einfachpendels erkennen, dass zunächst eine Datenakquise stattfinden muss, bevor die Fouriertransformation durchgeführt werden kann. Daraufhin pendelt sich der Fehler  $\nu_2$  in derselben Größenordnung ein. Gleiches lässt sich im Auszug aus der Schätzung der Parameter erkennen, welche zum gleichen Wert konvergieren, nachdem die automatisch gewählte Bibliothek in Betrieb ist. Ausführliche Details, auch zur Bibliothek, sind in der Dissertation [17] zu finden. Mittels dieser Ergänzung kann eine höhere Regelgüte erzielt werden, indem im geschlossenen Regelkreis eine aktive Störkompensation, bei der Informationen auf Basis der Approximation  $\hat{\rho}$  ins Modell zurückgeführt werden, durchgeführt wird. Die Erweiterung einer automatisierten Bibliothekswahl ist somit eine Ergänzung des augmentierten Beobachters und stellt eine hilfreiche Maßnahme dar, welche als Werkzeug in der intelligenten Fehlererkennung und Störkompensation gewinnbringend eingesetzt werden kann. Diese Grundidee einer automatisierten Gestaltung der Bibliothek basierend auf erhobenen Daten wird daher im folgenden Abschnitt 5.3 für die Modelladaption aufgegriffen.

Das Konzept des augmentierten Beobachters ist **übertragbar** auf andere Beobachertypen, sofern die Voraussetzungen erfüllt sind. In der Kombination mit robusten Sliding-Mode-Beobachtern kann eine Bibliothek durch Fourieranalyse **automatisch** bestimmt werden und **Vorteile** im Bereich Störidentifikation und -kompensation bieten.



**Abbildung 5.15** Verlauf der Schaltfunktion  $\nu_2$ , welche die Modelldiskrepanz  $\Delta f$  widerspiegelt, und Auszug aus den Parameterverläufen  $\hat{\theta}$ , vgl. [17], [209]

## 5.3 Automatische Modellaktualisierung

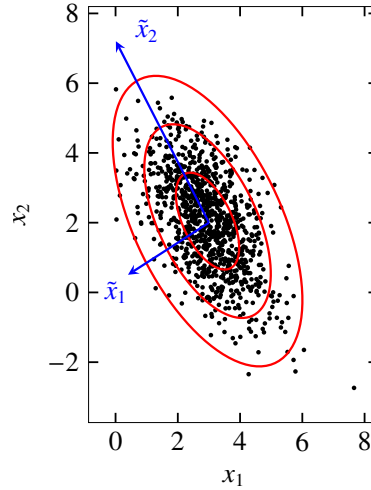
Um die Informationen, die durch den augmentierten Beobachter gewonnen werden, auch langfristig nutzen zu können, beispielsweise um die Modellgüte zu verbessern, ist eine sukzessive Modellaktualisierung wünschenswert. Durch eine Modelladaptation wird die Modellgüte in der Regel erhöht, sodass eine Verwendung des verbesserten Modells auch für weitere Entwurfsschritte wie die Auslegung eines Reglers vorteilhaft ist. Eine automatische Modellaktualisierung erfordert jedoch zunächst eine statistisch basierte Merkmalsanalyse und -extraktion, welche im folgenden Abschnitt thematisiert werden. Anschließend kann auf Basis der extrahierten Merkmale eine Modellaktualisierung erfolgen. Zudem wird ein Konzept zur fortwährenden Modelladaptation vorgestellt, welches besonders gewinnbringend im Fall auftretender Systemveränderungen oder sich schnell verändernder Systeme ist, vgl. [17].

### 5.3.1 Merkmalsextraktion

Grundlage der Modellaktualisierung bildet die *Hauptkomponentenanalyse (PCA)*<sup>8</sup>, welche auf der Singulärwertzerlegung basiert und auf die Arbeiten von [210], [211] zurückgeht. Da durch den augmentierten Beobachter Schätzwerte des Parametervektors  $\theta_k$  für vergangene Zeitschritte  $k = 1, \dots, N$  vorliegen, können diese in einer Zeitreihenmatrix  $\Theta \in \mathbb{R}^{n_\theta \times N}$  gesammelt werden. Ziel der PCA ist eine Koordinatentransformation der vorliegenden Daten  $\Theta$  in ein Koordinatensystem, welches diese besser als das vorherige, meist kartesische Koordinatensystem darstellen kann. Diese Grundidee ist in der Abbildung 5.16 beispielhaft für zwei Dimensionen visualisiert.

<sup>8</sup>Im Englischen ist diese als *Principal Component Analysis (PCA)* bekannt. Die Abkürzung PCA wird zunehmend auch im Deutschen verwendet, sodass diese im Folgenden genutzt wird.

Da sich die Daten, illustriert durch die Punkte, anhand ihrer Varianz charakterisieren lassen, befindet sich in der Abbildung 5.16 die durch die PCA transformierte erste Achse entlang der größten Varianz der Daten, während die zweite Achse orthogonal zu dieser steht und die zweitgrößte Varianz in den Daten beschreibt.



**Abbildung 5.16** Koordinatentransformation durch die PCA anhand eines zweidimensionalen Beispiels: Die roten Ellipsen stellen jeweils die einfache, doppelte und dreifache Standardabweichung dar, die blauen Achsen beschreiben das durch die PCA gefundene Koordinatensystem, vgl. [17], [28].

Die Datenmatrix  $\Theta$  besteht jedoch nicht nur aus zwei, sondern aus  $n_\theta$  verschiedenen Merkmalen, die in  $N$  Beobachtungen, z. B. durch Experimente, erfasst worden sind. Wenn die Daten eine unterschiedliche Skalierung aufweisen, ist eine Vorverarbeitung dieser unerlässlich, vgl. Abschnitt 2.2. Erfolgt diese nicht, verzerrt die PCA die tatsächlich zugrunde liegenden Informationen (vgl. [15, Abbildungen 2.1 und 2.2]). Aufgrund der Sensitivität der PCA werden die Daten zunächst standardisiert, woraufhin die standardisierte Datenmatrix  $\Theta_S$  in einer Kovarianzmatrix

$$P_\Theta = \Theta_S \Theta_S^T \quad (5.39)$$

angeordnet wird. Anschließend werden deren Eigenwerte  $\lambda_i$ <sup>9</sup> und Eigenvektoren  $\mathbf{v}_i$  mit  $i = 1, \dots, n_\theta$  bestimmt. Da die Eigenwerte von  $P_\Theta$  eine enge Verwandtschaft zu den Singulärwerten von  $\Theta$  aufweisen, können diese mithilfe der Singulärwertzerlegung effizient berechnet werden (vgl. [15], [28]). Die dazugehörigen Eigenvektoren  $\mathbf{v}_i$  sind dabei orthonormal zueinander. In der Regel wird die PCA nicht nur zur Analyse der Daten genutzt, sondern auch zur Dimensionsreduktion. Dies gelingt unter der Prämisse, dass manche der Achsen mit geringer Varianz lediglich Rauschen enthalten, woraufhin die darin enthaltenen Informationen vernachlässigt werden können. Zum Zweck der Modellaktualisierung muss daher entschieden werden, welche der  $n_\theta$  Eigenwerte beibehalten oder für die Darstellung der gesammelten Daten vernachlässigt werden können. Da die Daten bereits vorverarbeitet worden sind, ist der Anteil eines Eigenwerts, die gesammelten Daten gut darstellen zu können, umso

<sup>9</sup>Im Kontrast zu vorigen Kapiteln bezeichnet das Symbol  $\lambda$  einen Eigenwert in diesem Abschnitt.

höher, je größer dieser ist. Die PCA liefert genau eine solche Sortierung der Eigenwerte  $\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_{n_\theta}$  und ihrer Eigenvektoren  $\mathbf{v}_1, \dots, \mathbf{v}_{n_\theta}$ . Zur Bestimmung, ab welchem Eigenwert  $\lambda_{i^*}$  mit  $1 \leq i^* \leq n_\theta$  eine Modellreduktion auf die ersten ein bis  $i^*$  Eigenwerte stattfindet, existieren überwiegend einfache, empirische Kriterien, beispielsweise die Kaiser-Methode, der Scree-Test oder das Verfahren der kumulativen Varianz [15], [212]. Letzteres wird im Folgenden genutzt und kennzeichnet sich durch eine Abschätzung basierend auf der Varianz der Daten  $\mathcal{Q}$ :

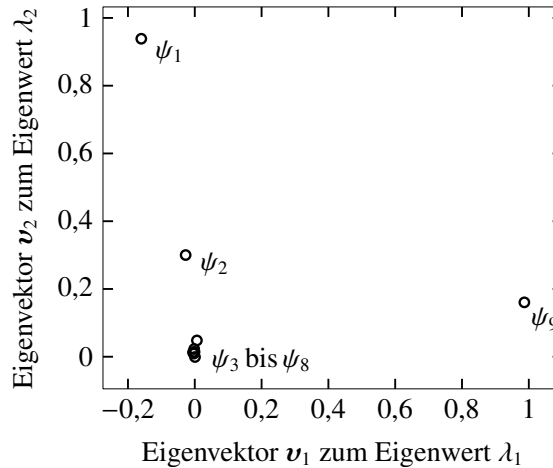
$$\mathcal{Q} = \frac{\sum_{l_1=1}^{i^*} \lambda_{l_1}}{\sum_{l_2=1}^{n_\theta} \lambda_{l_2}} \cdot 100. \quad (5.40)$$

Um die Fähigkeit, die gesammelten Daten adäquat zu repräsentieren, quantitativ zu bewerten, wird in der Gleichung (5.40) zunächst ein Quotient bestehend aus der Summe der ersten  $i^*$  Eigenwerte bezogen auf die Summe aller Eigenwerte gebildet. Dieser Quotient wird daraufhin prozentual ausgewertet und stellt somit die prozentuale Varianz der Daten dar [15], [212]. Indem ein Prozentsatz vorgegeben wird, den das reduzierte Modell bezogen auf die Daten in jedem Fall darstellen können muss, wird die kumulative Varianz  $\mathcal{Q}$  mit diesem Wert verglichen. Schließlich werden die  $i^*$  Eigenwerte behalten, die mit ihrer kumulativen Varianz  $\mathcal{Q}$  genau die vorgegebene Grenze überschreiten. In der Regel wird ein Prozentsatz zwischen 70% und 90% gewählt [15].

Nachdem die relevanten Eigenwerte bestimmt worden sind, können anschließend die dazugehörigen Eigenvektoren genutzt werden, um auf die einzelnen Terme  $\psi_i$  zu schließen. Dazu kann geometrisch argumentiert werden: Für jeden Eigenwert zeigt der betragsmäßig größte Eintrag des Eigenvektors in genau die Richtung des Elements, der den Eigenwert am meisten dominiert. In der Abbildung 5.17 sind zur Illustration des Vorgehens die Elemente der Eigenvektoren  $\mathbf{v}_1, \mathbf{v}_2$  der beiden größten Eigenwerte  $\lambda_1, \lambda_2$  in der  $x$ - bzw.  $y$ -Achse dargestellt. Die Elemente der Eigenvektoren  $v_{1,i}$  bzw.  $v_{2,i}$  sind in der Abbildung durch den Index des Bibliotheksterms  $\psi_i$  dargestellt, sodass z. B. die Position von  $\psi_9$  in der Abbildung durch das neunte Element von  $\mathbf{v}_1$  und das neunte Element von  $\mathbf{v}_2$  bestimmt wird. Somit visualisiert die Abbildung 5.17 die Beiträge der einzelnen Bibliotheksterme zum jeweiligen Eigenwert. Für den größten Eigenwert  $\lambda_1$ , der in diesem Beispiel 81,69% Anteil an der Varianz besitzt, weist  $\psi_9$  den stärksten Beitrag auf, da das neunte Element von  $\mathbf{v}_1$  betragsmäßig am größten ist. Dies ist in der Abbildung daran zu erkennen, dass alle anderen Terme bzgl. der  $x$ -Achse nahe Null positioniert sind. Für den zweitgrößten Eigenwert, der 12,2% Anteil an der Varianz besitzt, kann anhand der  $y$ -Achse abgelesen werden, dass sowohl das erste als auch das zweite Element des Eigenvektors  $\mathbf{v}_2$  einen Einfluss besitzen. Da das erste Element, erkennbar an der Position des Terms  $\psi_1$ , jedoch einen betragsmäßig größeren Einfluss aufweist, wird dieses als dominant bestimmt und deshalb auf den Term  $\psi_1$  zurückgeführt. Alle weiteren Terme besitzen keinen Einfluss auf diesen Eigenwert, da deren Beträge nahe Null sind, was durch die Überlagerung der Terme um Null dargestellt ist. Folglich wird zu jedem der dominanten Eigenwerte  $\lambda_i$  der Eigenvektor  $\mathbf{v}_i$  bzgl. seines betragsmäßig größten Elements ausgewertet. Dieses Element  $v_{l^*}$  mit  $1 \leq l^* \leq n_\theta$  ist dann die Referenz für den Bibliotheksterm  $\psi_{l^*}$ , der den stärksten Einfluss auf den Eigenwert  $\lambda_i$  aufweist.

Exemplarisch wird diese Merkmalsanalyse auf das Beispiel des Duffing-Oszillators



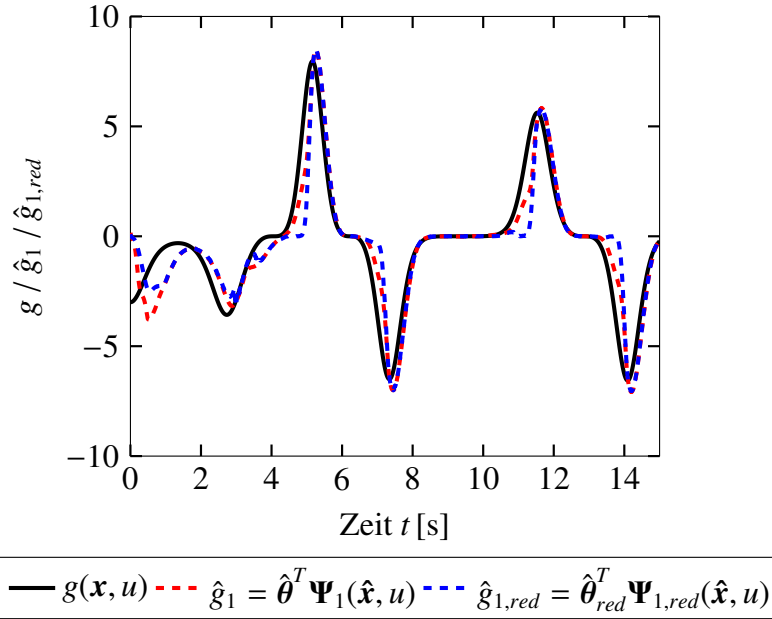


**Abbildung 5.17** Die ersten beiden Eigenwerte  $\lambda_1, \lambda_2$  decken zusammen mehr als 93% der Varianz ab. Die Elemente der dazugehörigen Eigenvektoren  $v_1$  und  $v_2$  sind in dieser Visualisierung in Abhängigkeit voneinander als Kreise dargestellt. Die Zahlen  $i$  stellen das jeweilige Element des Eigenvektors  $v_1$  bzw.  $v_2$  dar und können infolgedessen auf die Bibliotheksterme  $\psi_i$  zurückgeführt werden, vgl. [17].

mit der Bibliothek  $\Psi_1$  angewendet, dessen Daten aus den Abbildungen 5.5 und 5.7 verwendet werden. Werden schließlich die beiden wichtigsten, dominanten Terme  $\psi_9$  und  $\psi_1$  genutzt, um die Modellungenauigkeit zu approximieren, kann diese reduzierte Identifikation  $\hat{g}_{1,red}$  zur vollständigen Linearkombination  $\hat{g}_1$  und zur Modellungenauigkeit  $g$  verglichen werden. In der Abbildung 5.18 ist dieser Vergleich dargestellt, welcher aufzeigt, dass aufgrund der Modellreduktion zwar geringfügig Informationen verloren gehen, die  $\hat{g}_1$  aufweist und welche sich daher als Abweichung bei  $\hat{g}_{1,red}$  bemerkbar machen, im Allgemeinen aber die Genauigkeit weiterhin hoch ist. Somit liefert die Modellreduktion durch die PCA die Identifikation relevanter Terme, die eine physikalisch-technische Interpretierbarkeit der Modellungenauigkeit ermöglichen.

### 5.3.2 Modellaktualisierung

Auf Basis der PCA stehen die relevanten, dominanten Bibliotheksterme  $\psi_i$  fest, welche die Modellungenauigkeiten  $g$  im vergangenen betrachteten Zeitraum am besten charakterisieren. Allerdings ist für die Aktualisierung des Modells die Kenntnis des jeweiligen dazugehörigen physikalischen Parameters erforderlich. Dieser kann u. U. zeitinvariant, aber auch zeitvariant vorliegen und muss durch eine nachfolgende Parameteridentifikation bestimmt werden. Anschließend kann das Modell um die identifizierte, parametrische Darstellung für die Modellungenauigkeit ergänzt werden. Allerdings ist es in manchen Situation erforderlich und vorteilhaft, dass eine fortwährende Identifikation und Modellaktualisierung durchgeführt wird. Die Abbildung 5.19 zeigt ein Flussdiagramm, welches den Ablauf einer solchen dauerhaften Modelladaption darstellt. Das Konzept besteht aus zwei Phasen, die abhängig von Kriterien durchlaufen werden. Die erste Phase umfasst die Datenakquise und die Überprüfung mithilfe verschiedener Kriterien, ob eine Aktualisierung notwendig ist. Diese Kriterien umfassen den Zeitpunkt  $T_0$ , welcher bei einem erstmaligen Be-



**Abbildung 5.18** Approximation der Modellungenauigkeit  $g$  durch die vollständige Linearkombination  $\hat{g}_1$  und durch die von der PCA reduzierte Linearkombination  $\hat{g}_{1,red}$ , vgl. [17], [197]

trieb des Systems das Ende der Einschwingphase bzw. des transienten Verhaltens markiert, sowie die Anzahl der gesammelten Parametersätze, da genügend aussagekräftige Zeitreihendaten gesammelt werden müssen, um eine Analyse durchführen zu können. Entscheidend ist zudem, ob eine Aktualisierung des Modells erforderlich ist. Dieses Merkmal stellt häufig der Modellfehler  $\Delta f$  dar, welcher beispielsweise anhand des Ausgangsfehlers evaluiert werden kann. Überschreitet dieser eine definierte Grenze  $\delta_f$ , beginnt die zweite Phase, welche in der Abbildung 5.19 dargestellt ist. Mithilfe der PCA und einer festgelegten kumulierten Varianz  $Q$  werden die dominanten Bibliotheksfunktion  $\psi_i$  des vergangenen, analysierten Zeitraums extrahiert, woraufhin die restlichen Einträge in der Bibliothek entfernt werden. Um eine weiterhin sich sukzessiv verbessernde Modellgüte zu ermöglichen, werden neue Bibliotheksterme basierend auf vorgefertigten Bibliotheken  $\boldsymbol{\Psi}_i^*$  gebildet. Diese Bibliotheken können beispielsweise thematisch geordnet sein und je nach Anwendungsfall trigonometrische Funktionen oder typische Reibelemente enthalten. Die Auswahl neuer Bibliotheksterme  $\psi_{j,new}$  erfolgt dabei randomisiert oder infolge einer weitergehenden Analyse. Letztere ist sinnvoll, wenn viele Eigenwerte gleich starke Beiträge zur Varianz liefern, sodass möglicherweise nicht alle dominanten Bibliotheksterme identifiziert wurden. Anwendungsbeispiele dieses Vorgehens sind in der Dissertation [17] zu finden.

Bei Systemveränderungen oder sich schnell verändernden Systemen ist eine **automatische Modelladaption** durch den augmentierten Beobachter mittels der **Hauptkomponentenanalyse** möglich. Dadurch ist eine fortwährende Option zur **sukzessiven Modellverbesserung** gegeben. Voraussetzungen sind weiterhin eine geeignet gewählte Bibliothek sowie ausreichend große Datenmengen und Analysezeiträume.

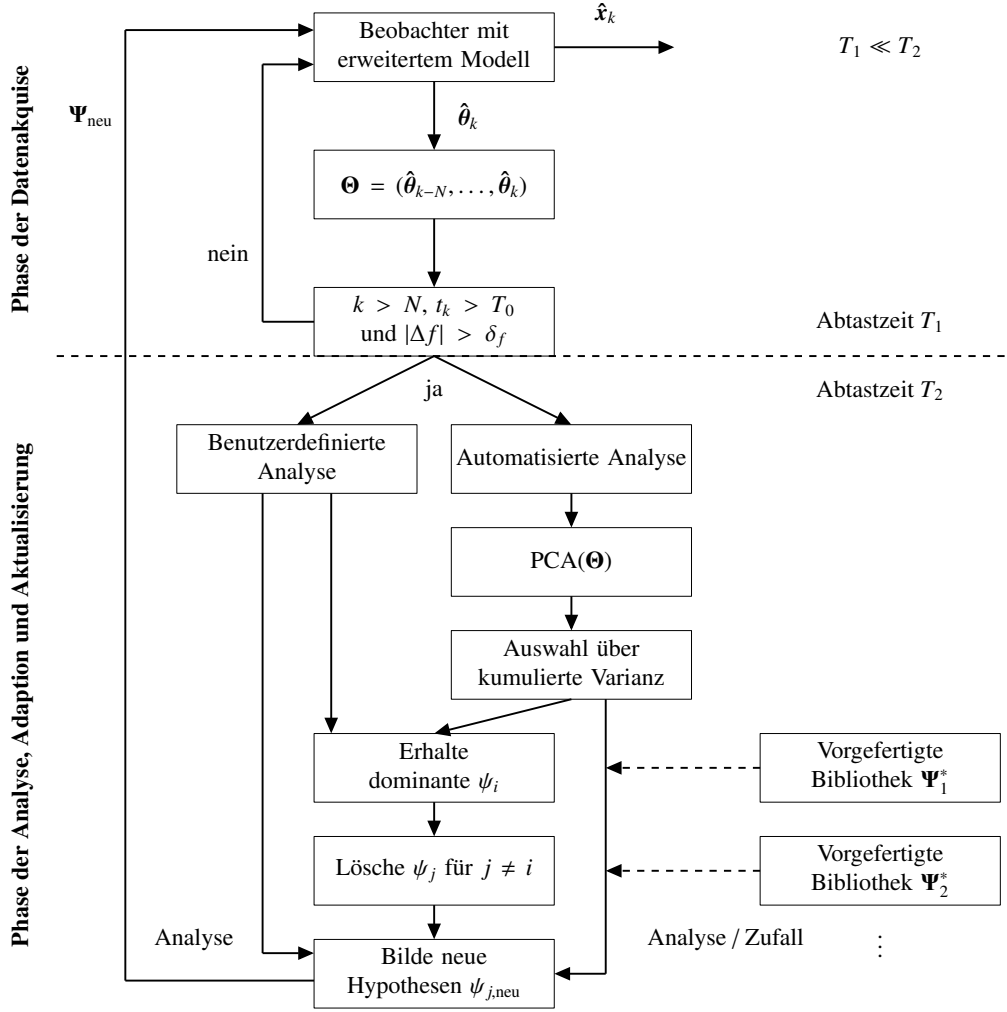


Abbildung 5.19 Konzept zur simultanen Schätzung und Modellaktualisierung, vgl. [17]

## 5.4 Koopman-basierter Beobachter mit Verwendung von EDMD

In [213] wird erstmals der modellbasierte Beobachterentwurf für ein Koopman-basiertes Streckenmodell in Anlehnung an lineare Luenberger-Beobachter beschrieben. Es wird angenommen, dass sich die Dynamik eines zugrunde liegenden autonomen Systems mittels  $N$  Koopman-Eigenfunktionen  $\Phi(\mathbf{x}) = [\varphi_1(\mathbf{x}), \dots, \varphi_N(\mathbf{x})]^\top$  und den dazugehörigen Koopman-Eigenwerten  $\lambda_1, \dots, \lambda_N$  durch

$$\dot{\Phi}(\mathbf{x}) = \mathcal{K}\Phi(\mathbf{x}) = \Lambda\Phi(\mathbf{x}) \quad (5.41)$$

und der  $n$ -dimensionale Systemzustand

$$\mathbf{x} = \sum_{j=1}^N \varphi_j(\mathbf{x}) \mathbf{v}_j^x = \mathbf{V}^x \Phi(\mathbf{x}) \quad \text{mit} \quad \mathbf{v}_j^x \in \mathbb{R}^n, \mathbf{V}^x = [\mathbf{v}_1^x, \dots, \mathbf{v}_N^x] \in \mathbb{R}^{n \times N} \quad (5.42)$$

sowie der  $q$ -dimensionale Systemausgang

$$\mathbf{y} = \sum_{j=1}^N \varphi_j(\mathbf{x}) \mathbf{v}_j^y = \mathbf{V}^y \Phi(\mathbf{x}) \quad \text{mit} \quad \mathbf{v}_j^y \in \mathbb{R}^q, \mathbf{V}^y = [\mathbf{v}_1^y, \dots, \mathbf{v}_N^y] \in \mathbb{R}^{q \times N} \quad (5.43)$$

als Linearkombinationen formulieren lassen<sup>10</sup>. Dann ergibt sich daraus die Koopman-Beobachternormalform

$$\Phi(\mathbf{x}_k) = \Lambda \Phi(\mathbf{x}_{k-1}), \quad (5.44a)$$

$$\mathbf{y}_k = \mathbf{V}^y \Phi(\mathbf{x}_k), \quad (5.44b)$$

$$\mathbf{x}_k = \mathbf{V}^x \Phi(\mathbf{x}_k), \quad (5.44c)$$

mit der ein Luenberger-Beobachter [24], [181]

$$\hat{\Phi}(\mathbf{x}_k) = \Lambda \hat{\Phi}(\mathbf{x}_{k-1}) + \mathbf{L}(\mathbf{y}_k - \hat{\mathbf{y}}_k), \quad (5.45a)$$

$$\hat{\mathbf{y}} = \mathbf{V}^y \hat{\Phi}(\mathbf{x}_k), \quad (5.45b)$$

$$\mathbf{x}_k = \mathbf{V}^x \hat{\Phi}(\mathbf{x}_k) \quad (5.45c)$$

entworfen werden kann. Wenn das Paar  $(\Lambda, \mathbf{V}^y)$  beobachtbar ist, kann eine Rückführmatrix  $\mathbf{L}$  bestimmt werden, sodass der Beobachter (5.45) konvergiert. Falls die Koopman-Eigenzerlegung (5.41)-(5.43) darüber hinaus endlich dimensional und exakt ist, ist die Koopman-Beobachternormalform (5.44) global gültig und damit der Beobachter (5.45) global konvergent. In [213] wird simulativ demonstriert, dass die resultierende Schätzgüte eines Koopman-basierten Kalman-Filters sowohl anhand eines einschlägigen Einführungsbeispiels, vgl. Gleichung (3.8), als auch am Van-der-Pol-Oszillator einem EKF überlegen ist. [214] erweitert den beschriebenen Ansatz auf eingangsaffine und bilineare Systeme und [215] erweitert den Ansatz auf Systeme mit Zustandsbeschränkungen.

In [216] wird die simulative Entwicklung eines Kalman-Filters unter Verwendung eines DMD-Modells zur Modellordnungsreduktion für Windturbinen beschrieben. Der Beitrag [217] beschreibt in Anlehnung an [213], [214] die Entwicklung eines robusten Koopman-basierten Kalman-Filters mit einem Maximum-Likelihood-Ansatz<sup>11</sup> für ein elektrisches Energiesystem und demonstrieren eine höhere Schätzgüte als bei der Verwendung eines EKFs. Der Beitrag [219] beschreibt einen neuartigen Ansatz für die Fehlererkennung bei supraleitenden Hochfrequenzkavitäten in einem Teilchenbeschleuniger mittels eines Koopman-basierten Kalman-Filters. Im Vergleich zu einem UKF ermöglicht der Koopman-basierte Kalman-Filter bei dieser Anwendung eine hohe Schätzgüte bei einer stark verringerten Rechenzeit. Der Beitrag [220] beschreibt den Entwurf eines Koopman-basierten EKFs für die Kraft- und Drehmomentschätzung eines weichen Medizinroboters. Hierfür wurde das EDMD-

<sup>10</sup>Hier wird der Einfachheit halber ausschließlich der Fall für reale Eigenwerte beschrieben; eine Erweiterung um komplex-konjugierte Eigenwerte findet sich in [213].

<sup>11</sup>Die Maximum-Likelihood-Methode [218] stammt aus der Statistik und konstruiert eine Schätzfunktion derart, dass für den unbekannten Parameter unter der Grundgesamtheit aller denkbaren Schätzwerte genau derjenige ausgewählt wird, bei dem die gezogene Stichprobe die maximale Eintrittswahrscheinlichkeit besitzt. Eine umfassende Beschreibung dieser Methode findet sich beispielsweise in [10].

Prädiktionsverfahren mit Korrektur, vgl. Unterabschnitt 3.1.2, aufgegriffen, um die Schätzgüte zu erhöhen.

**Beispielhafte Anwendung am Schlagmechanismus des Golfroboters** Bei dem Schlagmechanismus des Golfroboters lässt sich der Zustand  $x_1$  (der Winkel der Abtriebswelle) messen, sodass für die Ausgangsgleichung gilt, vgl. Gleichung (6.11),

$$y = \mathbf{c}^\top \mathbf{x} \quad \text{mit} \quad \mathbf{c}^\top = [1, 0]. \quad (5.46)$$

Daher ist es erforderlich, dass der Zustand  $x_2$  mittels eines Beobachters geschätzt wird. Basierend auf dem EDMD-Modell der Dynamik lässt sich ein Luenberger-Beobachter [24], [181] für den Schlagmechanismus entwerfen, der sich durch

$$\dot{\hat{\Psi}}(\mathbf{x}) = \left( \mathbf{K} - \tilde{\mathbf{l}}\tilde{\mathbf{c}}^\top \right) \hat{\Psi}(\mathbf{x}) + \mathbf{b}u + \tilde{\mathbf{l}}y \quad (5.47)$$

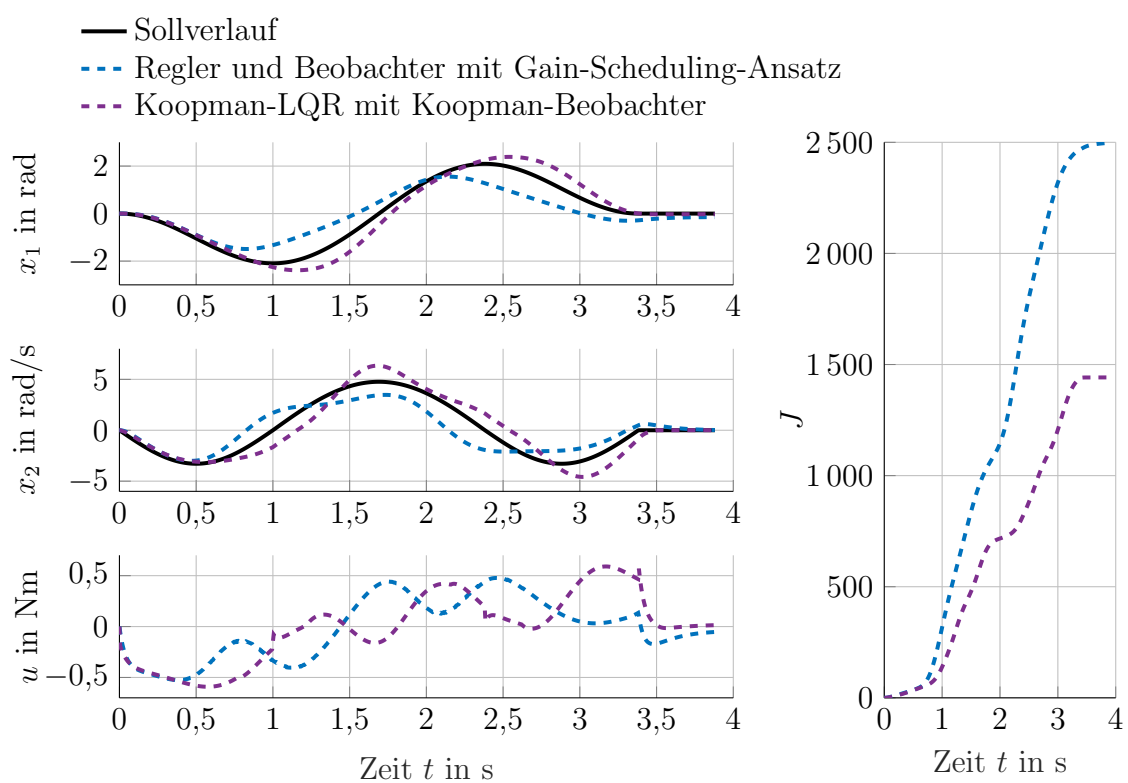
beschreiben lässt. Hierbei ist der EDMD-Ausgangsvektor durch

$$\tilde{\mathbf{c}} = [1, 0, 0, 0] \quad (5.48)$$

gegeben und die Eigenwerte des Beobachters, d. h. der Matrix  $\mathbf{K} - \tilde{\mathbf{l}}\tilde{\mathbf{c}}^\top$ , werden doppelt so groß wie die Eigenwerte des geschlossenen Regelkreises gewählt [221].

In der Abbildung 5.20 wird beispielhaft die resultierende Regelgüte (6.16)-(6.17) des EDMD-basierten Beobachters mit einem EDMD-basierten LQ-Regler der des physikalisch motivierten Entwurfs, vgl. Beginn dieses Kapitels, gegenübergestellt. Bei letzterem werden der Regler und Beobachter durch einen Gain-Scheduling-Ansatz bestimmt. Für die Solltrajektorie mit einem Ausholwinkel von  $120^\circ$  erreicht der Koopman-Ansatz eine deutlich höhere Regelgüte. Diese Erkenntnis ist nach umfassenden Analysen der vorherigen Kapitel erwartungsgemäß und lässt sich darauf zurückführen, dass das EDMD-Modell eine höhere Modellgenauigkeit aufweist, vgl. Unterabschnitt 3.1.2, und somit auch eine überlegene Regelgüte, vgl. Abschnitt 4.1, als der Gain-Scheduling-Ansatz bietet, der auf einer Linearisierung des nichtlinearen physikalischen Modells basiert. Aufgrund der hohen Schätzgüte und der geradlinigen Anwendbarkeit lässt sich die Aussage treffen, dass das EDMD-Modell eine hohe regelungstechnische Verwertbarkeit auch hinsichtlich des Beobachterentwurfs aufweist.

EDMD-Modelle lassen sich aufgrund der linearen Systemstruktur geradlinig für den Entwurf eines Luenberger-Beobachters nutzen. Am Beispiel des Schlagmechanismus des Golfroboters wurde mittels numerischer Simulationen die Erkenntnis gewonnen, dass der EDMD-basierte Entwurfsansatz bezüglich der resultierenden Regelgüte einem physikalisch motivierten Ansatz mit klassischer Linearisierung überlegen sind.



**Abbildung 5.20** Resultierende simulierte Regelgüte unter Einsatz von Beobachtern.

## 6 Demonstratoren

Ein wichtiges Projektziel von DART war es, die entwickelten hybriden Methoden nicht nur an akademischen und simulativen Beispielen zu testen, sondern auch an realen regelungstechnischen Prüfständen. Die Erfahrung zeigt, dass es einen großen Unterschied macht, Methoden auch real zu testen, denn viele Störeffekte der realen Welt wie Rauschen und Verzögerungen werden in Simulationen häufig nicht berücksichtigt bzw. können vorab nicht abgeschätzt werden. Im Rahmen des Projekts wurde der Golfroboter zum einen selbst weiterentwickelt und zum anderen der Schlagmechanismus für viele regelungstechnische Experimente verwendet. Die Funktionsweise des Golfroboters wird in Abschnitt 6.1 vorgestellt, sodass neben den einzelnen Experimenten zu den hybriden Methoden auch der Gesamtentwurf des automatisierten Golfspiels erläutert wird. Der selbstbalancierende Würfel ist ein während der Projektlaufzeit entwickelter regelungstechnischer Prüfstand, der genutzt wird, um ebenfalls für Experimente der hybriden Methoden zur Verfügung zu stehen. Der Würfel ist in der Lage sich mit Hilfe der Beschleunigung und anschließend abruptem Abbremsen von Schwungrädern auf eine Kante oder Ecke aufzustellen und kann dort stabilisiert werden. Als sehr anschaulicher regelungstechnischer Demonstrator wird der Würfel auch über die Projektlaufzeit hinaus genutzt werden. In Abschnitt 6.2 wird der Aufbau und die Funktionsweise erläutert und es werden einige Experimente im Rahmen der Tests der entwickelten hybriden Methoden beschrieben.

### 6.1 Golfroboter

Der Golfroboter ist ein selbstlernendes mechatronisches System, welches datengetriebene und physikalische Methoden kombiniert, um das Putten eines Golfballes von einem beliebigen Punkt des Greens autonom zu lernen. Hierfür wird ein Kamerasystem mit Bilderkennung verwendet und außerdem ein künstliches neuronales Netz gelernt, welches den Geschwindigkeitsvektor des Schlages vorhersagen kann, um ein erfolgreiches Einlochen sicher zu stellen. Um die Anzahl der zeitaufwändigen Interaktionen mit dem realen System zu minimieren, wird das neuronale Netz vortrainiert. Hierfür wird ein physikalisches Modell ausgewertet, das die Dynamik des Golfballs auf dem Green nachbildet. Die Approximation des Greens erfolgt ebenfalls datengetrieben. Dieses mechatronische Anwendungsbeispiel ist somit in der Lage die synergetische Kombination von datengetriebenen und physikalisch basierten Methoden zu demonstrieren.

Autonome Roboter werden voraussichtlich schon in naher Zukunft viele Menschen in alltäglichen Tätigkeiten unterstützen können, zum Beispiel in der Versorgung von alten oder körperlich eingeschränkten Personen, die gehoben oder bewegt werden müssen. Hierfür ist ein umsichtiges Handeln des autonomen Roboters un-



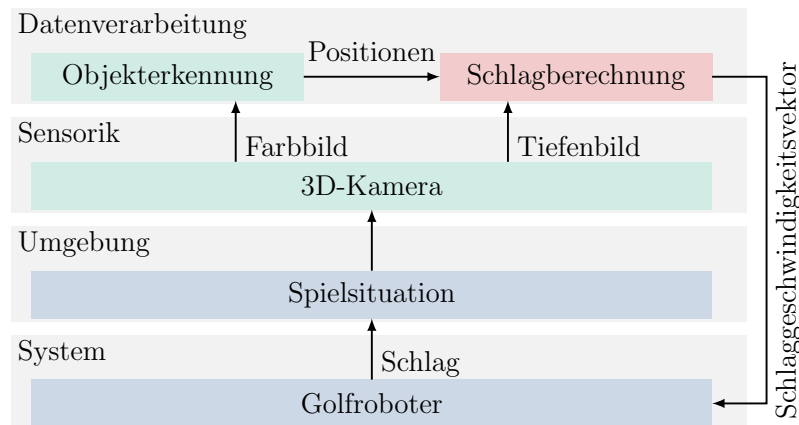
**Abbildung 6.1** *Golfi* ist ein selbstlernender Golfroboter, der in der Lage ist autonom zu putten. *Golfi* dient als Demonstrator für die Anwendung von datengetriebenen Methoden in der Regelungstechnik.

erlässlich. Diese und andere Aspekte von autonomen Robotern behandeln wir in unserer Forschungsarbeit und wollen sie am Beispiel des autonomen Golfroboters weiterentwickeln. Das Szenario, in dem sich der Golfroboter bewegt ist es, im Bereich des Greens den Golfball in das Loch zu putten. Um diese Herausforderung zu bewältigen, benötigt ein System, welches autonom arbeiten soll, eine genaue Umgebungserkennung und eine präzise Ansteuerung, denn selbst für geübte Golfspieler ist das Putten anspruchsvoll und nicht immer erfolgreich. Um diesen Anforderungen gerecht zu werden, kombinieren wir leistungsfähige datengetriebene Methoden mit etablierten physikalischen Methoden aus der Regelungstechnik. Ein hybrider Ansatz erscheint hier äußerst vorteilhaft, um die Vorteile aus beiden Bereichen optimal zu nutzen.

Im Golfsport gibt es einige Roboter, die eine Vielzahl unterschiedlicher Aufgaben übernehmen. Ein Bereich ist die Unterstützung der Spieler bei der korrekten Ausführung der Schläge. Beispielsweise optimiert der Roboter in [222] den Schwung der Golf spielenden Person, indem er ihren Arm direkt führt. Eine weitere verbreitete Anwendung von Golfrobotern ist das Testen des Zubehörs. Beispiele aus [223], [224], [225], [226], [227], [228], [229] zeigen Roboter, die Golfschläger und Golfbälle mit vielen verschiedenen Schlägen testen, wobei die Grundkonstruktion meist aus einem rotierenden Roboterarm besteht. Der dritte Anwendungsbereich besteht in der Imitation von menschlichen Schlägen. Der Roboter *ROB-OT* [230] ist in der Lage ein komplettes Golfspiel zu spielen und dient daher als ein Demonstrationsobjekt für Golfer, aber auch Unterhaltungszwecken. Obwohl dieser Roboter in der Lage ist, sich auf dem gesamten Green zu bewegen, wird ein\*e Golfexpert\*in für die Bedienung während des gesamten Spiels benötigt.

Unser Golfroboter, wie er in der Abbildung 6.1 zu sehen ist, hat das Ziel vollständig autonom zu putten. Das bedeutet, dass er den Ball von einer beliebigen Position auf einem unbekannten Green mit einem einzigen Schlag einlochen soll, wobei der Ball nicht durch die Luft fliegen soll. Um dies zu erreichen wird eine Kombination aus klassischer Regelungstechnik und datengetriebenen Techniken, also maschinell-





**Abbildung 6.2** Systemarchitektur des Golfroboters: Das Gesamtproblem ist in Teilaufgaben unterteilt, wobei unterschiedliche Ansätze verfolgt werden: datengetrieben, physikalisch basiert oder auch hybrid. Der Startpunkt im Prozess ist eine gegebene Situation in einem Golfspiel.

lem Lernen, verwendet. Die Positionierungs- und Schlageinheit können recht einfach physikalisch modelliert werden, sodass sich hierfür klassische Optimierungsmethoden nutzen lassen. Im Gegensatz dazu sind die Analyse der Spielsituation und die Bestimmung einer optimalen Schlagrichtung und Schlaggeschwindigkeit herausfordernde Probleme, die nicht einfach mit Hilfe von physikalischen Gesetzen gelöst werden können. Daher strukturieren wir die Gesamtaufgabe des autonomen Golfspiels in unterschiedliche Teilaufgaben, wie es in der Abbildung 6.2 aufgezeigt wird. Die Komplexität nimmt dabei von unten nach oben zu und damit auch der Anteil der verwendeten datengetriebenen Ansätze. Ganz unten wird das mechatronische Grundsystem des Golfroboters beschrieben, welches dafür zuständig ist, den Roboter auf dem Green zu bewegen und den Schlag mit einer vorgegebenen Richtung und Schlaggeschwindigkeit ausführt. Die aktuelle Situation des Golfspiels wird definiert durch die Positionen von Golfer, dem Ball, dem Loch und auch der Beschaffenheit und Größe des Greens und diese Gegebenheiten werden durch eine 3D-Kamera aufgenommen. Auf der obersten Ebene werden zum einen datengetriebene Methoden des maschinellen Lernens verwendet, um Objekte zu detektieren und zum anderen eine synergetische Kombination von datengetriebenen und physikbasierten Methoden, um den Geschwindigkeitsvektor für das Einlochen des Balles zu berechnen. Mit diesem Vorgehen ist es möglich zunächst ein künstliches neuronales Netz vorzutrainieren, indem simulativ erzeugte Trainingsschläge verwendet werden, die ein physikalisches Modell der Balldynamik auf einem vorgegebenen Green nutzen. Anschließend wird dieses neuronale Netz mittels Trainingsschlägen des realen Golfroboters nachtrainiert. Dieses Vorgehen reduziert die Anzahl der zeitaufwendigen Interaktionen mit dem realen System signifikant und erzeugt daher ein besseres Ergebnis durch eine zielgerichtete und sinnvolle Kombination von datengetriebenen und physik-basierten Methoden.

### 6.1.1 Mechatronischer Entwurf des Golfroboters

Das mechatronische System des Golfroboters muss einen vorgegebenen Schlaggeschwindigkeitsvektor realisieren, d. h. den Ball in eine bestimmte Richtung mit ei-

ner spezifischen Geschwindigkeit schlagen. Diese Aufgabe wird in zwei Teilaufgaben unterteilt.

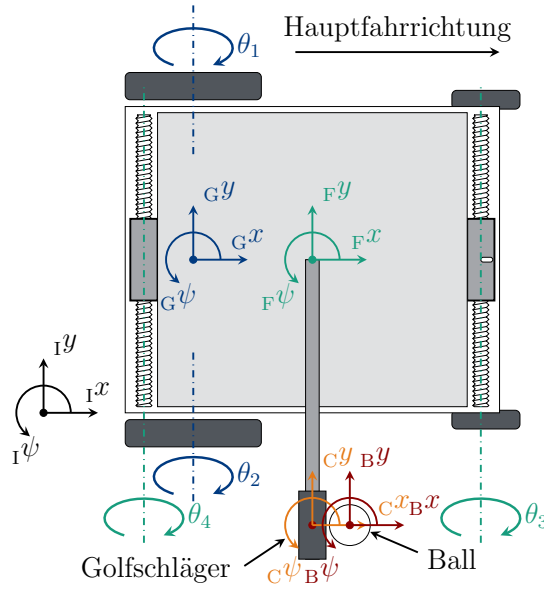
1. Die Positionierungseinheit muss den Golfroboter so auf dem Green platzieren, dass der Schläger sich direkt neben dem Ball befindet und dabei in eine bestimmte Richtung zeigt und
2. der Schlagmechanismus muss den Ball so treffen, dass er mit einer spezifischen Anfangsgeschwindigkeit zu rollen beginnt.

Im Folgenden werden diese Aufgaben detailliert beschrieben.

### Positionierungseinheit

Die Positionierungseinheit des Golfroboters besteht aus einer Fahreinheit und einer zusätzlichen Feinpositioniereinheit, wie in der Abbildung 6.3 schematisch dargestellt ist. Da der Golfplatz durch die an der Decke montierte Kamera aus der Vogelperspektive betrachtet wird, vereinfacht sich die Positionierungsaufgabe zu einem ebenen Problem, bei dem die  $x$ -Achse, die  $y$ -Achse und die Rotation  $\Psi$  in der Ebene ausreichen, um die Positionierung vollständig zu beschreiben. Das Inertialkoordinatensystem I, das dem Kamerakoordinatensystem entspricht, liegt ungefähr im Mittelpunkt des Greens. Das körperfeste Koordinatensystem G liegt zwischen den beiden hinteren Rädern und ist so ausgerichtet, dass die  ${}_Gx$ -Achse in Fahrtrichtung und die  ${}_Gy$ -Achse nach links zeigt.

Die Fahreinheit, die ein Chassis mit zwei angetriebenen JMC-Servomotoren umfasst, die durch eine Arduino-Plattform angesteuert werden, sowie zwei frei drehbar gelagerten Rädern, realisiert eine Translation entlang der  ${}_Gx$ -Achse und eine Drehung um den Winkel  ${}_G\Psi$ . Obwohl der Golfroboter theoretisch durch diese beiden Freiheitsgrade jede beliebige Pose einnehmen kann, ist die Translation entlang der  ${}_Gy$ -Achse schwierig zu realisieren. Daher wird zusätzlich eine Feinpositioniereinheit verwendet, die durch das Koordinatensystem F beschrieben wird. Dieses befindet sich zwischen dem Chassis und dem Schlaggerät und basiert auf zwei Spindeln, die von Joy-IT-Schrittmotoren mit Leadshine DM 542-Treibern angetrieben werden. Die Feinpositioniereinheit ermöglicht eine kleine, aber weitaus präzisere Bewegung des Roboters, weil hier der Einfluss der Traktion auf dem Boden eliminiert wird. Bezüglich des Koordinatensystems F ist eine translatorische Bewegung entlang der  ${}_Fy$ -Achse und eine Rotation um den Winkel  ${}_F\Psi$  im Bereich von  $\pm 18^\circ$  möglich. Die Räder und Spindeln können direkte translatorische Verfahrbefehle erhalten, wobei angenommen wird, dass die Verfahrbefehle aufgrund der Eigenschaft der Schrittmotoren ideal realisiert werden. Außerdem wird das Koordinatensystem C eingeführt, das sich im Zentrum des Schlägers befindet, sowie das Ball-Koordinatensystem B, das sich im Zentrum des Balls befindet und die gleiche Ausrichtung wie I hat. Das Ziel der Positionierung besteht darin, die  ${}_Cx$ -Achse mit der Richtung des gewünschten Schlaggeschwindigkeitsvektors  ${}_I\mathbf{v}_s$  auszurichten und den Ursprung des Koordinatensystems C mit einem Offset von 3 cm in negativer  ${}_I\mathbf{v}_s$ -Richtung ab dem Ursprung von B zu platzieren. In den bisherigen Tests wurde der Einfachheit halber nur die Fahreinheit und noch nicht die Feinpositioniereinheit verwendet, wodurch dennoch ausreichende Ergebnisse erzielt werden konnten.



**Abbildung 6.3** Die Positionierungseinheit, hier in Draufsicht, besteht aus einer Fahreinheit und einer zusätzlichen Feinpositioniereinheit. Die Fahreinheit umfasst zwei hintere Räder, die einzeln angetrieben werden (in blau) mit den Drehmomenten  $\theta_1$  und  $\theta_2$ , wobei das Gewicht des Roboters von den beiden frei drehbar gelagerten Rädern vorne mit getragen wird. Die Antriebseinheit ermöglicht es dem Roboter, sich um den Winkel  ${}_G\Psi$  zu drehen und Bewegungen entlang der  ${}_Gx$ -Achse auszuführen. Die Feinpositioniereinheit, die auf zwei einzeln angetriebenen Spindeln (in grün) basiert, verfügt über ein Langloch auf dem rechten Wagen, das es dem Roboter ermöglicht, sich um kleine Winkel  ${}_F\Psi$  zu drehen, indem die Drehmomente  $\theta_3$  und  $\theta_4$  in entgegengesetzte Richtungen angetrieben werden, sowie Bewegungen entlang der  ${}_Fy$ -Achse.

Die Pose des Golfschlägers wird durch

$${}^I\mathbf{g}_C = [{}^Ix_C, {}^Iy_C, {}^Iz_C]^\top \quad (6.1)$$

bezeichnet. Das Ziel der Ansteuerung ist es, dass die tatsächliche Endpose des Golfschlägers  ${}^I\mathbf{g}_{C,e}$  der gewünschten Endpose des Golfschlägers  ${}^I\mathbf{g}_{C,d}$  entspricht, die von der Ballposition und dem Schlaggeschwindigkeitsvektor abhängt. Eine Trajektorie zwischen einem Start- und einem Zielpunkt unter Berücksichtigung von Hindernissen zu ermitteln ist in der Literatur ein bekanntes Problem, für welches unterschiedliche Ansätze bestehen. Am Golfroboter wird die Fahrtrajektorie über ein Optimierungsproblem ermittelt. Der Ansatz bietet die Möglichkeit das Verhalten des Roboters zu bestrafen, wodurch sich u. A. die Kollision mit dem Ball und bestehenden Hügeln vermeiden lässt. Die Ansteuerung für die Schrittmotoren wird daher durch die Minimierung der Kostenfunktion

$$J_p(\Theta) = \phi_Z({}^I\mathbf{g}_{C,e}) + \phi_S(\Theta) + \phi_G({}^I\mathbf{g}_C) + \phi_B({}^I\mathbf{g}_C) + \phi_H({}^I\mathbf{g}_C) \quad (6.2)$$

berechnet, wobei

$$\Theta = [\Theta(1), \dots, \Theta(N)] \quad \text{mit} \quad \Theta(i) = \begin{bmatrix} \theta_1(i) \\ \theta_2(i) \end{bmatrix} \quad \forall i = 1, \dots, N \quad (6.3)$$

die Ansteuerungssequenz durch die Stellgrößen  $\theta_1$  und  $\theta_2$  wiedergeben.  $\theta_1$  und  $\theta_2$  entsprechen dabei den Drehgeschwindigkeiten beider Räder, die für ein Manöver  $i$  konstant gewählt werden. Die gesamte Fahrtrajektorie setzt sich aus  $N$  Dreh- und Fahrmanövern zusammen, wobei die Annahme, dass es sich bei  $i = 1$  um eine Drehung handelt, die benötigte Anzahl an Manövern  $N$  in den meisten Fällen verringert. In Abbildung 6.4 ist eine beispielhafte valide Trajektorie aus  $N = 6$  Manövern dargestellt. Die Kostenfunktion enthält mehrere Strafterme, worüber sich das Verhalten des Roboters beeinflussen lässt. Der Term

$$\phi_Z(\mathbf{I}\mathbf{g}_{C,e}) = (\mathbf{I}\mathbf{g}_{C,d} - \mathbf{I}\mathbf{g}_{C,e})^\top \mathbf{Q}_p (\mathbf{I}\mathbf{g}_{C,d} - \mathbf{I}\mathbf{g}_{C,e}) \quad (6.4)$$

sorgt für das korrekte Erreichen der Zielposition. Ferner fließt ein hoher Verbrauch der Stellgrößenenergie über

$$\phi_S(\Theta) = \sum_{i=1}^N \Theta^\top(i) \mathbf{R}_p \Theta(i) \quad (6.5)$$

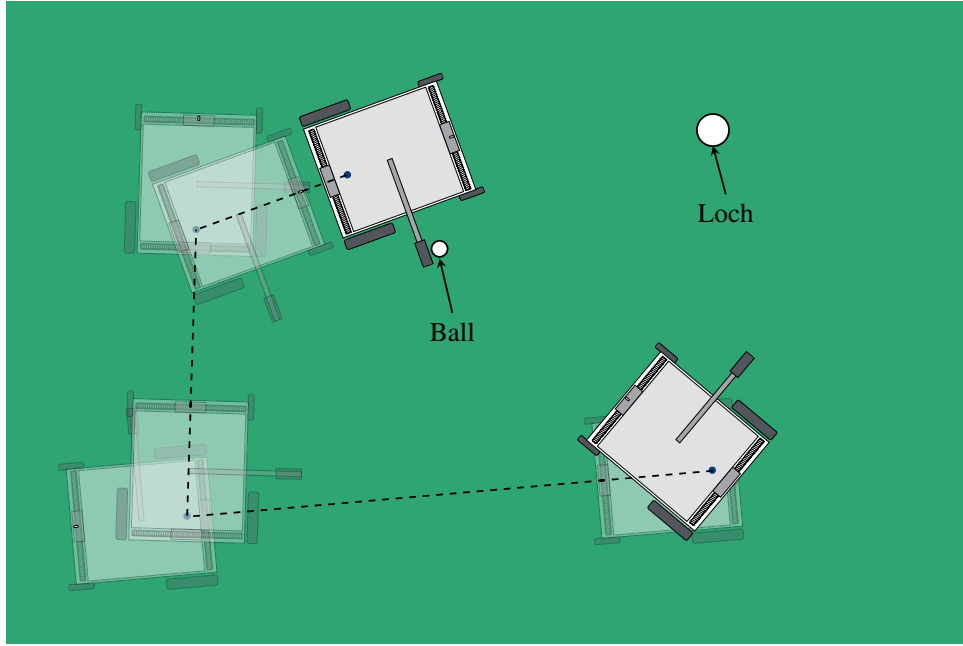
in die Kostenberechnung ein, wodurch unnötige Fahrwege reduziert werden. Beide Strafterme  $\phi_Z(\mathbf{I}\mathbf{g}_{C,e})$  und  $\phi_S(\Theta)$  lassen sich über die Gewichtungsmatrizen  $\mathbf{Q}_p$  und  $\mathbf{R}_p$  in Relation zueinander setzen. Die weiteren Terme

$$\phi_G(\mathbf{I}\mathbf{g}_C) = \begin{cases} 0, & \text{für } \sum_{i=1}^N \mathbf{I}\mathbf{g}_{C,i} \in \mathbf{B}_G, \\ \sum_{i=1}^N \mathbf{S}_G, & \text{für } \sum_{i=1}^N \mathbf{I}\mathbf{g}_{C,i} \notin \mathbf{B}_G, \end{cases} \quad (6.6a)$$

$$\phi_B(\mathbf{I}\mathbf{g}_C) = \begin{cases} 0, & \text{für } \sum_{i=1}^N \mathbf{I}\mathbf{g}_{C,i} \notin \mathbf{B}_B, \\ \sum_{i=1}^N \mathbf{S}_B, & \text{für } \sum_{i=1}^N \mathbf{I}\mathbf{g}_{C,i} \in \mathbf{B}_B, \end{cases} \quad (6.6b)$$

$$\phi_H(\mathbf{I}\mathbf{g}_C) = \begin{cases} 0, & \text{für } \sum_{i=1}^N \mathbf{I}\mathbf{g}_{C,i} \notin \mathbf{B}_H, \\ \sum_{i=1}^N \mathbf{S}_H, & \text{für } \sum_{i=1}^N \mathbf{I}\mathbf{g}_{C,i} \in \mathbf{B}_H \end{cases} \quad (6.6c)$$

lassen sich das Verlassen des Greens ( $\phi_G$ ) und eine Kollision mit dem Ball ( $\phi_B$ ) oder einem Hügel ( $\phi_H$ ) an jeder Position  $\mathbf{g}_C$  eines potentiellen Fahrwegs vermeiden. Sowohl das Verlassen des Greens als auch die Kollisionsvermeidung werden durch eine Abfrage über Beinhaltung der aktuellen Position des Roboters  $\mathbf{I}\mathbf{g}_{C,i}$  in der zugehörigen Menge aller validen Zustände für das Green  $\mathbf{B}_G$ , Ball  $\mathbf{B}_B$  und Hügel  $\mathbf{B}_H$  realisiert. Im Fall einer Kollision oder beim Verlassen des Greens erhöht sich die Summe in Gleichung (6.2) um einen konstanten Wert  $\mathbf{S}_G$ ,  $\mathbf{S}_B$  oder  $\mathbf{S}_H$ . Da eine Trajektorie mit auftretender Kollision bzw. Verlassen des zulässigen Bewegungsraums ausgeschlossen werden soll, empfiehlt es sich die Strafterme ausreichend hoch zu wählen in Relation zu den Strafen für weite Fahrmanöver  $\phi_Z(\mathbf{I}\mathbf{g}_{C,e})$  oder ungenaue Positionierung  $\phi_S(\Theta)$ . Für das Lösen des Optimierungsproblems wird ein Partikelschwarmalgorithmus verwendet, wofür die Anzahl der Dreh- und Fahrmanöver  $N$  vorgegeben werden muss. Bei steigendem  $N$  wird der Suchraum für valide Trajektorien exponentiell größer, wodurch sich die Rechendauer für den Algorithmus erhöht. Das Ziel ist es somit  $N$  hinreichend niedrig zu wählen. Um diesen Parameter passend abzuschätzen werden alternative Fahrtrajektorien verwendet, die jedoch nicht die gleichen Eigenschaften wie im vorgestellten Optimierungsproblem



**Abbildung 6.4** Der Roboter in der Draufsicht an mehreren Positionen auf einem beispielhaften Green ohne Hügel abgebildet. Damit der Roboter bei der Positionierung den Ball nicht verschiebt, fährt er ihn von hinten mit einer finalen translatorischen Bewegung an. Bei diesem letzten Fahrmanöver ist der Roboter bereits korrekt ausgerichtet. In dem Beispiel werden je drei Dreh- und Fahrmanöver ( $N = 6$ ) durchgeführt, wobei an den Drehpunkten die Positionen vor und nach der Drehung durch eine transparente Draufsicht des Roboters gekennzeichnet sind.

berücksichtigen. Unter der Verwendung des Rapidly Exploring Random Tree\* Algorithmus (RRT\*)[231], [232] wird ein solcher alternativer Weg für den Roboter für die Parameterschätzung geplant. Hierfür wird der gefundene Weg über einen zusätzlichen Algorithmus gekürzt, sodass unter Vermeidung der Hügel, die Anzahl der Dreh- und Fahrmanöver minimiert wird. In der Anwendung am Golfroboter resultiert dieses Vorgehen meist in je zwei bis vier notwendigen Dreh- und translatorischen Fahrmanövern.

Im Anschluss an die Bewegungsplanung erfolgt das Abfahren der Fahrtrajektorie, die durch das Lösen des Optimierungsproblems automatisiert ermittelt wurde. Analog zum Regelungsentwurf für einen kontrollierten Schlag in Abschnitt 6.1.1 – ohne Verwendung eines Beobachters – wurde dieser Ansatz ebenfalls für kontrolliertes Fahren verwendet. Der Roboter wurde in dem Zuge auf seine Hinterachse reduziert, woraus sich die Bewegungsgleichungen

$$\dot{\mathbf{g}}_{C,x} = \frac{r_{\text{Rad}}}{2} (\theta_1 + \theta_2) \cos(\mathbf{g}_{C,\psi}) = \frac{r_{\text{Rad}}}{2} (u_1 + u_2) \cos(x_3), \quad (6.7a)$$

$$\dot{\mathbf{g}}_{C,y} = \frac{r_{\text{Rad}}}{2} (\theta_1 + \theta_2) \sin(\mathbf{g}_{C,\psi}) = \frac{r_{\text{Rad}}}{2} (u_1 + u_2) \sin(x_3), \quad (6.7b)$$

$$\dot{\mathbf{g}}_{C,\psi} = \frac{r_{\text{Rad}}}{l_{\text{Achse}}} (\theta_2 - \theta_1) = \frac{r_{\text{Rad}}}{l_{\text{Achse}}} (u_2 - u_1) \quad (6.7c)$$

für das kinematische Modell ergeben, wobei die folgenden Ersatzgrößen für Zustands-

und Eingangsvektoren zur Vereinfachung gewählt werden

$$\mathbf{I}\dot{\mathbf{g}}_C = \begin{bmatrix} \mathbf{I}\dot{\mathbf{g}}_{C,x} \\ \mathbf{I}\dot{\mathbf{g}}_{C,y} \\ \mathbf{I}\dot{\mathbf{g}}_{C,\psi} \end{bmatrix} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}, \quad \mathbf{\Theta} = \begin{bmatrix} \theta_1 \\ \theta_2 \end{bmatrix} = \begin{bmatrix} u_1 \\ u_2 \end{bmatrix}. \quad (6.8)$$

In diesen Bewegungsgleichungen entspricht  $r_{\text{Rad}}$  dem Radius der baugleichen Hinterräder und  $l_{\text{Achse}}$  der Länge der Hinterachse, die beide Räder miteinander verbindet. In Gleichung (6.7) wird die gemittelte Drehgeschwindigkeit  $v = \frac{r_{\text{Rad}}}{2}(\theta_1 + \theta_2)$  in x- und y-Richtung aufgeteilt. Die Ausrichtung  $x_3$  ändert sich dabei nach Gleichung (6.7c) ebenfalls abhängig der Drehgeschwindigkeiten. Für das Beispiel  $u_1 = -u_2 > 0$  würde der Roboter, dargestellt in Abbildung 6.3, sich im Uhrzeigersinn um den Punkt  $\mathbf{I}\mathbf{g}_C$  drehen. Die Gleichungen Gleichung (6.7) können zusammengefasst werden durch die Beschreibung  $\mathbf{I}\dot{\mathbf{g}}_C = \mathbf{A}_P \mathbf{g}_C + \mathbf{B}_P \mathbf{\Theta}$ . Auf Grundlage eines linearisierten Modells, wofür der Gain-Scheduling-Ansatz verwendet wird, lässt sich der Entwurf eines Riccati-Reglers durchführen. Der Zustand  $\mathbf{I}\dot{\mathbf{g}}_C$  wird über die Auswertung des Kamerasignals gemessen, wie es in Unterabschnitt 6.1.2 erläutert ist.

## Schlagmechanismus

Die Regelungsaufgabe des Schlagmechanismus besteht darin, dass der Schläger den Ball mit einer präzise geregelten translatorischen Geschwindigkeit  $\|\mathbf{v}_s\|$  schlägt. Dies entspricht einer rotatorischen Geschwindigkeit von  $\dot{\varphi}_s = \frac{\|\mathbf{v}_s\|}{h}$ , wobei  $h$  der Abstand von der Rotationsachse der Abtriebswelle zum Schlagpunkt des Schlägers ist, vgl. Tabelle 6.1. Die Schlaggeschwindigkeit  $\dot{\varphi}_s$  soll genau dann erreicht werden, wenn der Schläger senkrecht nach unten steht, d. h. einen Winkel von  $\varphi = 0$  rad aufweist. Basierend auf diesen Anforderungen wurde eine Schar von Solltrajektorien  $\mathbf{w} = [\varphi, \dot{\varphi}]^\top$  in Abhängigkeit der gewünschten Schlaggeschwindigkeit  $\dot{\varphi}_s$  hergeleitet. In Anlehnung an eine authentische Golfschwungbewegung bestehen diese jeweils aus den drei Phasen *Ausholen*, *Schlagen*, *Rückholen*, vgl. Abbildung 6.5, wobei die frei wählbaren Parameter  $\varphi_l$  und  $T_l$  den Winkel und die Dauer des Aushol- und Rückholvorgangs charakterisieren, vgl. Tabelle 6.1. Die abschnittsweise definierten Funktionen  $\varphi, \dot{\varphi} : \mathbb{R} \rightarrow \mathbb{R}$  ergeben sich durch

$$\varphi(t) = \begin{cases} f_1(t), & 0 < t \leq T_l \\ f_2(t), & T_l < t \leq \left(T_l + \frac{\varphi_l \pi h}{\|\mathbf{v}_s\|}\right) \\ f_3(t), & \left(T_l + \frac{\varphi_l \pi h}{\|\mathbf{v}_s\|}\right) < t \leq \left(2T_l + \frac{\varphi_l \pi h}{\|\mathbf{v}_s\|}\right) \\ 0, & \text{sonst,} \end{cases} \quad (6.9a)$$

$$\dot{\varphi}(t) = \begin{cases} f_4(t), & 0 < t \leq T_l \\ f_5(t), & T_l < t \leq \left(T_l + \frac{\varphi_l \pi h}{\|\mathbf{v}_s\|}\right) \\ f_6(t), & \left(T_l + \frac{\varphi_l \pi h}{\|\mathbf{v}_s\|}\right) < t \leq \left(2T_l + \frac{\varphi_l \pi h}{\|\mathbf{v}_s\|}\right) \\ 0, & \text{sonst} \end{cases} \quad (6.9b)$$

mit

$$f_1(t) = \frac{\varphi_l}{2} \left( \cos \left( \frac{\pi}{T_l} t \right) - 1 \right), \quad (6.10a)$$

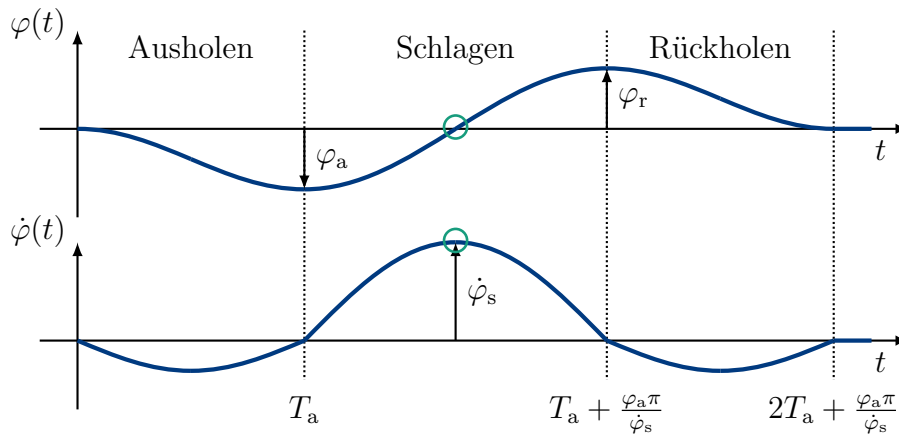
$$f_2(t) = -\varphi_l \cos \left( \frac{\|v_s\|}{\varphi_l h} (t - T_l) \right), \quad (6.10b)$$

$$f_3(t) = \frac{\varphi_l}{2} \left( \cos \left( \frac{\pi}{T_l} \left( t - T_l - \frac{\varphi_l \pi h}{\|v_s\|} \right) \right) + 1 \right), \quad (6.10c)$$

$$f_4(t) = -\frac{\varphi_l \pi}{2T_l} \sin \left( \frac{\pi}{T_l} t \right), \quad (6.10d)$$

$$f_5(t) = \frac{\|v_s\|}{h} \sin \left( \frac{\|v_s\|}{\varphi_l h} (t - T_l) \right), \quad (6.10e)$$

$$f_6(t) = -\frac{\varphi_l \pi}{2T_l} \sin \left( \frac{\pi}{T_l} \left( t - T_l - \frac{\varphi_l \pi h}{\|v_s\|} \right) \right). \quad (6.10f)$$



**Abbildung 6.5** Die Solltrajektorien für den Winkel  $\varphi$  und die Winkelgeschwindigkeit  $\dot{\varphi}$  bestehen aus den drei Phasen *Ausholen*, *Schlagen*, *Rückholen* und werden durch stückweise definierte Funktionen beschrieben, vgl. Gleichung (6.9). Die gewünschte Schlaggeschwindigkeit  $\dot{\varphi}_s$  wird erreicht, wenn der Winkel  $\varphi = 0$  aufweist, hier eingezeichnet als grüner Kreis.

**Tabelle 6.1** Parameter zur Beschreibung der Regelungsaufgabe des Golfroboters.

Symbol	Physikalischer Parameter
$T_l = T_r$	Ausholdauer bzw. Rückholdauer
$\varphi_l = -\varphi_r$	Ausholwinkel bzw. Rückholwinkel
$h$	Abstand von der Rotationsachse der Abtriebswelle zum Schlagpunkt des Schlägers

Die untere Welle wird durch einen drehmomentgeregelten Elektromotor<sup>1</sup> angetrieben und an der oberen Welle ist der Golfschläger mit einem Putting-Schlägerkopf montiert, vgl. Abbildung 6.1. Um die Sicherheit zu gewährleisten, darf der Schläger

<sup>1</sup>Beckhoff AM8042 [233].

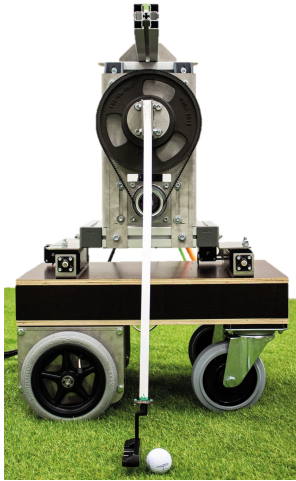
keine Überschlge ausfhren. Unter der Annahme, dass beide Getriebewellen zu einem starren Krper zusammengefasst sind<sup>2</sup>, ergibt sich durch physikalische berlegungen ein vereinfachtes dynamisches Zustandsraummodell des Schlagmechanismus

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} x_2 \\ \frac{-mga \sin x_1 - M_d(\mathbf{x}) + 4u}{J} \end{bmatrix}, \quad (6.11a)$$

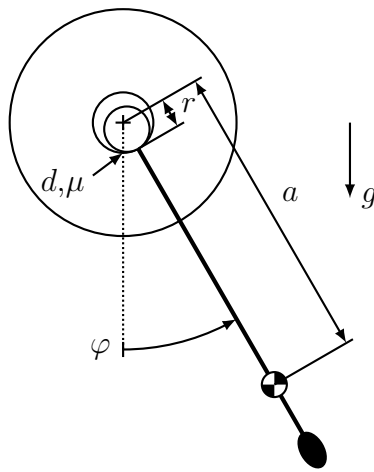
$$y = x_1, \quad (6.11b)$$

$$\text{mit } M_d(\mathbf{x}) = dx_2 + r\mu \operatorname{sgn} x_2 |mx_2^2 a + mg \cos x_1|. \quad (6.11c)$$

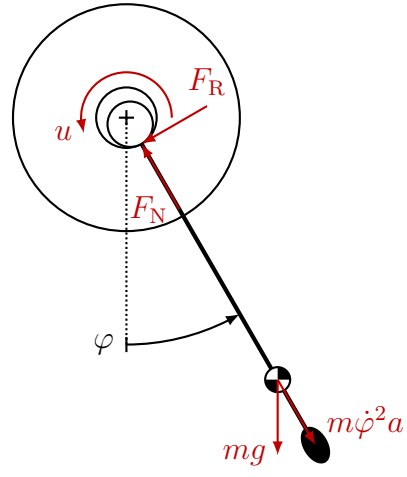
Der Zustandsvektor  $\mathbf{x} = [\varphi, \dot{\varphi}]^\top$  enthlt den Winkel  $\varphi$  und die Winkelgeschwindigkeit  $\dot{\varphi}$  der (oberen) Abtriebswelle. Die untere stabile Ruhelage des Schlgers entspricht der Nulllage  $\varphi = 0$ . Das nichtlineare Dmpfungsmoment  $M_d : \mathbb{R}^2 \rightarrow \mathbb{R}$  bildet die statische und die dynamische Reibung der Drehgelenke ab, vgl. Abbildung 6.6(c). Die physikalischen Parameter finden sich in der Tabelle 6.2 und in der Abbildung 6.6(b).



(a) Schlagmechanismus



(b) Ersatzbild



(c) Freischnitt

**Abbildung 6.6** Physikalisch motivierte Modellbildung des Schlagmechanismus. Es wird vereinfachend angenommen, dass beide Getriebewellen zu einem starren Krper zusammengefasst sind und daher nur die obere Zahnriemenscheibe betrachtet. In Anlehnung an die whrend des Projekts bearbeitete studentische Arbeit [234].

Die Regelungsstrategie des Schlagmechanismus basiert auf einer linearen Zwei-Freiheitsgrade-Struktur [24] unter Verwendung eines Gain-Scheduling-Ansatzes [21], wie in der Abbildung 6.7 dargestellt ist. Das bedeutet, dass die Regelstrecke (6.11) fr verschiedene Arbeitspunkte  $\varphi_{R_i} \in [-\pi, \pi]$  mit einem Inkrement von 0,01 rad linearisiert wird. Dadurch entsteht eine Reihe von linearen Teilmodellen

$$\dot{\mathbf{x}} = \mathbf{A}_{R_i} \mathbf{x} + \mathbf{b}u, \quad (6.12a)$$

$$y = \mathbf{c}^\top \mathbf{x} \quad (6.12b)$$

<sup>2</sup>Die Getriebebersetzung ist in der Bewegungsdifferentialgleichung (6.11) durch den Vorfaktor 4 beim Steuereingang bercksichtigt.



**Tabelle 6.2** Physikalische Parameter des Schlagmechanismus des Golfroboters.

Symbol	Physikalischer Parameter	Wert
$m$	Masse des Golfschlägers	0,5241 kg
$J$	Rotationsträgheit des Golfschlägers	0,1445 kg/m <sup>2</sup>
$g$	Gravitationsbeschleunigung	9,81 m/s <sup>2</sup>
$a$	Abstand von der Rotationsachse der Abtriebswelle zum Massenschwerpunkt des Golfschlägers	0,4702 m
$d$	Dynamischer Reibungskoeffizient	0,0132 kgm <sup>2</sup> /s
$r$	Radius von der Rotationsachse der Abtriebswelle zum Reibungspunkt	0,0245 m
$\mu$	Statischer Reibungskoeffizient	1,5136

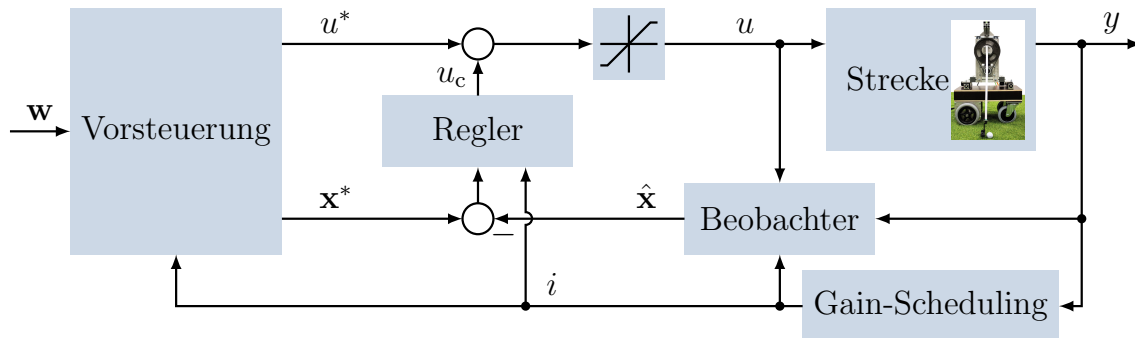
mit

$$\mathbf{A}_{R_i} = \begin{bmatrix} 0 & 1 \\ -\frac{m_c g a}{J} \cos(\varphi_{R_i}) & -\frac{d}{J} \end{bmatrix}, \quad \mathbf{b} = \begin{bmatrix} 0 \\ \frac{a}{J} \end{bmatrix}, \quad \mathbf{c}^\top = [1, 0]. \quad (6.13)$$

Während des Betriebs wird der Regler verwendet, dessen zugehöriges Streckenmodell am besten mit der aktuellen Situation übereinstimmt, d. h.

$$i = \arg \min_i |x_1 - \varphi_{R_i}|. \quad (6.14)$$

Die Gain-Scheduling-Variable  $i$  bestimmt sowohl die Vorsteuerung und den Regler als auch den für die Zustandsregelung erforderlichen Beobachter.

**Abbildung 6.7** Am Prüfstand wird eine lineare Zwei-Freiheitsgrade-Regelungsstruktur mit einem Gain-Scheduling-Ansatz verwendet. In Anlehnung an [235].

Die linearen Zustandsrückführungen

$$u_c = \mathbf{k}_{R_i}^\top (\mathbf{x}^* - \hat{\mathbf{x}}) \quad (6.15)$$

werden als zeitinvariante Riccati-Regler entworfen. Hierfür wird die quadratische Kostenfunktion

$$J = \frac{1}{2} \int_0^\infty \mathbf{x}^\top(t) \mathbf{Q} \mathbf{x}(t) + R u^2(t) dt \quad (6.16)$$

mit

$$\mathbf{Q} = \begin{bmatrix} 5 & 0 \\ 0 & 1 \end{bmatrix}, \quad R = 1 \quad (6.17)$$

minimiert, die die simultane Berücksichtigung des zeitlichen Verlaufs der Zustandsgrößen und der aufgebrachten Steuerenergie ermöglicht<sup>3</sup>.

Die Vorsteuerungsmatrizen, die zur Berechnung der Sollverläufe

$$\mathbf{x}^* = \mathbf{F}_x \mathbf{w}, \quad u^* = \mathbf{f}_{u_{R_i}}^\top \mathbf{w} \quad (6.18)$$

erforderlich sind, ergeben sich durch die Betrachtung des eingeschwungenen Zustands [24] zu

$$\mathbf{F}_x = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \quad \mathbf{f}_{u_{R_i}}^\top = -\mathbf{b}^+ \mathbf{A}_{R_i}. \quad (6.19)$$

Das resultierende Regelgesetz lautet

$$u = u^* + u_c, \quad (6.20)$$

wobei aufgrund von Stellgrößenbeschränkungen zu beachten ist, dass die Stellgröße  $u$  einer Sättigung unterliegt, vgl. Abbildung 6.7.

Für jedes linearisierte System wird ein Luenberger-Beobachter [24], [181] entworfen, dessen Dynamik durch

$$\dot{\hat{\mathbf{x}}} = (\mathbf{A}_{R_i} - \mathbf{l}_{R_i} \mathbf{c}^\top) \hat{\mathbf{x}} + \mathbf{b}u + \mathbf{l}_{R_i} y \quad (6.21)$$

beschrieben wird. Dabei werden die Eigenwerte des Beobachters, d. h. der Matrix  $\mathbf{A}_{R_i} - \mathbf{l}_{R_i} \mathbf{c}^\top$  so gewählt, dass sie doppelt so groß sind wie die Eigenwerte des geschlossenen Regelkreises [221].

### 6.1.2 Bildverarbeitung

Für ein erfolgreiches Putten muss die Informationsverarbeitung mittels geeigneter Sensorik alle erforderlichen Informationen über die Spielsituation erfassen. Dazu ist an der Decke des Labors eine Stereokamera<sup>4</sup> befestigt, die die Spielfläche aus der Vogelperspektive aufnimmt. Das Farbbild wird für die Detektion der Spielobjekte verwendet. Im Gegensatz dazu liefert das Tiefenbild die absoluten Positionen der detektierten Spielobjekte und dient darüber hinaus zur Approximation des Höhenprofils der Spielfläche. Im Folgenden werden die Elemente der Bildverarbeitung detailliert erklärt.

Die Größe und Position der Spielfläche sowie die Position des Lochs werden als konstante Werte festgelegt, weil die Kameraposition nicht verändert wird. Zur einfachen Detektion der Position und Orientierung des Golfroboters besitzt dieser zwei farbige Kreise auf der Oberseite, vgl. Abbildung 6.13, deren Positionen sich anhand des gewünschten RGB-Farbcodes bestimmen lassen. Es ist wichtig, dass die Farben einen möglichst hohen Kontrast zueinander aufweisen, damit sie gut unterschieden werden können. Durch eine Minimierung der gemittelten Abweichung des

<sup>3</sup>Eine ausführliche Beschreibung dieses Ansatzes findet sich in [24].

<sup>4</sup>Bei der Kamera handelt es sich um eine *Kinect für Windows v2*.

RGB-Werts eines Pixels zum gewünschten RGB-Wert, lassen sich eine Vielzahl an Pixeln im Farbbild bestimmen, worüber die Mittelpunkte der farbigen Kreise definiert werden. Für eine schnellere Datenverarbeitung kann das untersuchte Farbbild zuerst in Graustufen transformiert und von einem statischen Graustufenbild ohne Roboter und Ball, subtrahiert werden. Es bleiben nur die Pixel übrig, auf denen eine Farbveränderung erkannt werden kann, die zur Definition von Bereichen im Bild verwendet werden können. Innerhalb dieser Bereiche erfolgt nun eine schnellere Objekterkennung. Unter der Annahme, den Ball als homogenen weißen Punkt zu betrachten lässt sich die Ballposition ebenfalls über dieses Verfahren bestimmen. Aufgrund dieser Annahme ist das Verfahren jedoch weniger robust, weshalb alternativ ein vortrainiertes Faltungsnetz<sup>5</sup> in Matlab genutzt werden kann. Auf Grundlage von 50 zuvor aufgenommenen Trainingsbildern erkennt das Netz den Ball zuverlässig. Dieses Verfahren ist rechenintensiver jedoch robuster, weshalb es sich für eine offline Detektierung eignet. Wenn der Ball bei einem geregelten Fahrmanöver ebenfalls erkannt werden soll (vgl. Abschnitt 6.1.1), wird eine schnelle Objekterkennung priorisiert. Das Tiefenbild liefert schließlich die absoluten Positionen der Spielobjekte im Inertialkoordinatensystem I.

Das Höhenprofil der Spielfläche wird anhand eines Tiefenbilds (das im Vorhinein ohne vorhandene Spielobjekte aufgenommen wird) durch eine zweidimensionale Spline-Funktion in Matlab approximiert. Hierfür werden zuerst mithilfe eines Hampel-Filters potentielle Ausreißer entfernt. Im Anschluss werden fehlende Daten extrapoliert und der gesamte Datensatz geglättet, sodass eine in  $1x$ - und  $1y$ -Richtung differenzierbare Funktion entsteht, in welcher das ortsabhängige Gefälle bestimmt werden kann. Dieses Gefälle ist entscheidend für die Berechnung der Hangabtriebskraft des Balls und somit für die Rolldynamik des Balls, vgl. Unterabschnitt 6.1.3.

### 6.1.3 Bestimmung eines optimalen Schlages

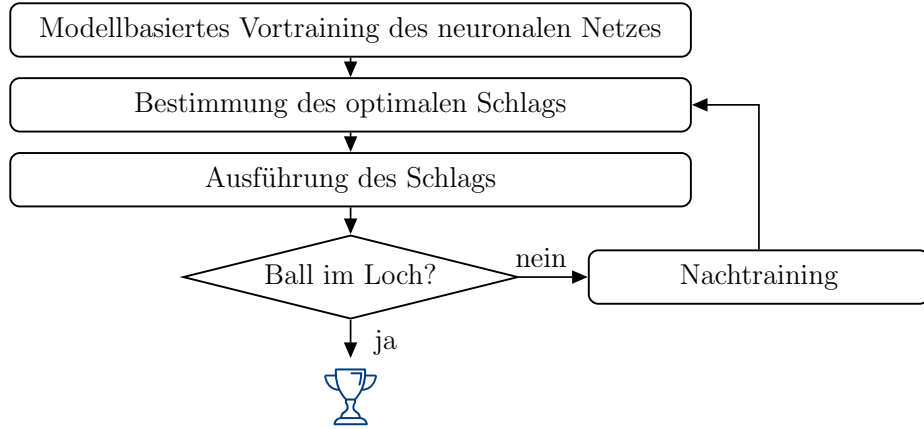
Das Kernelement bei der Bestimmung eines optimalen Schlags bildet ein neuronales Netz, das die Rolldynamik des Balls repräsentiert. Übergeordnet wird zur Bestimmung eines optimalen Schlags für ein erfolgreiches Putten die folgende Strategie verfolgt, vgl. Abbildung 6.8:

1. Vortrainieren des neuronalen Netzes für ein gegebenes Höhenprofil der Spielfläche anhand simulativ erzeugter Trainingsschläge, vgl. Abschnitt 6.1.3.
2. Bestimmung des optimalen Schlags anhand des neuronalen Netzes, vgl. Abschnitt 6.1.3.
3. Ausführung des Schlags auf der realen Spielfläche. Für den Fall, dass der Ball nicht unmittelbar in das Loch rollt, kann ausgehend von dieser Situation erneut eine Objektdetektion mit anschließender Berechnung und Ausführung eines optimalen Schlags durchgeführt werden.
4. Nachtrainieren des neuronalen Netzes. Dieser Schritt wurde bisher noch nicht am Golfroboter realisiert. Es ist jedoch denkbar, einen gescheiterten Schlag als

---

<sup>5</sup>Im Detail handelt es sich um ein sogenanntes *Faster R-CNN* [236].

weiteren Trainingsdatenpunkt zurückzuführen und somit die Genauigkeit des neuronalen Netzes zu erhöhen.



**Abbildung 6.8** Übergeordnete Strategie zur Bestimmung eines optimalen Schlags.

Im Folgenden werden die simulative Erzeugung der Trainingsdaten basierend auf einem Modell der Rolldynamik des Balles und die Bestimmung des optimalen Schlages mittels des neuronalen Netzes beschrieben.

### Erzeugung der simulativen Trainingsdaten

Die Erzeugung der Trainingsdaten erfolgt durch numerische Simulation eines physikalischen Modells der Rolldynamik des Balles. Diese wird maßgeblich durch die initiale Rollgeschwindigkeit, das Höhenprofil der Spielfläche sowie den Rollwiderstand des Balles auf der Spielfläche bestimmt.

Die Berechnung der Hangabtriebskräfte in  $x$ - und  $y$ -Richtung erfolgt mittels der lokalen Winkel

$$\alpha_x = \arctan \left( \frac{\partial f_{\text{green}}}{\partial x} \right), \quad (6.22a)$$

$$\alpha_y = \arctan \left( \frac{\partial f_{\text{green}}}{\partial y} \right) \quad (6.22b)$$

der Oberfläche  $f_{\text{green}}(x, y)$  der Spielfläche, vgl. Abbildung 6.9(a). Die Rollwiderstandskraft

$$F_r = m_b g \mu_b \cos \alpha_x \cos \alpha_y \quad (6.23)$$

wird als konstant angenommen [237] und wirkt stets parallel zur Oberfläche in entgegengesetzter Richtung

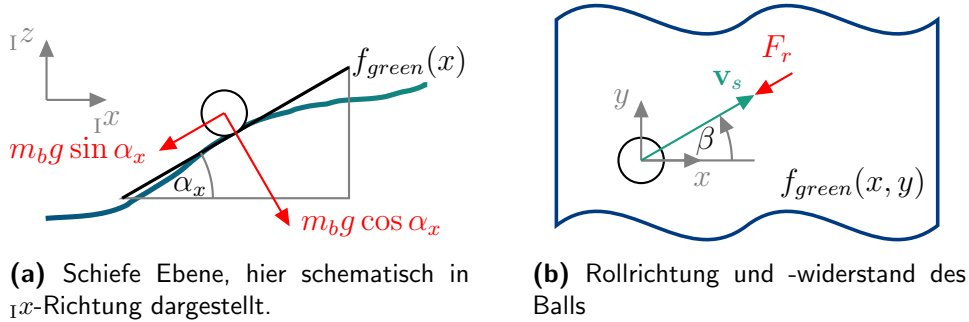
$$\beta = \arctan \left( \frac{\dot{y}}{\dot{x}} \right) \quad (6.24)$$

der Rollrichtung, vgl. Abbildung 6.9(b). Die resultierenden Bewegungsdifferentialgleichungen sind gegeben durch

$$m_b \ddot{x} = -m_b g \sin \alpha_x - F_r |\cos \beta| \text{sgn}(\dot{x}), \quad (6.25a)$$

$$m_b \ddot{y} = -m_b g \sin \alpha_y - F_r |\sin \beta| \text{sgn}(\dot{y}), \quad (6.25b)$$

sodass sich die korrespondierende Zustandsraumdarstellung mit dem Zustandsvektor  ${}_I\mathbf{q} = [{}_Ix, {}_Iy, {}_I\dot{x}, {}_I\dot{y}]^\top$  ergibt.



**Abbildung 6.9** Das physikalische Modell der Rolldynamik des Balls basiert auf der Betrachtung mittels schiefer Ebenen.

Für das Vortraining werden zufällige Schläge (mit unterschiedlichen Startpositionen und -geschwindigkeiten) anhand des physikalischen Modells (6.25) mit den Parametern aus Tabelle 6.3 simuliert<sup>6</sup>. Die Start- und Endpositionen des Balls sind durch  ${}_I\mathbf{q}_0 = [{}_Ix_0, {}_Iy_0, {}_I\dot{x}_0, {}_I\dot{y}_0]$  bzw.  ${}_I\mathbf{q}_e = [{}_Ix_e, {}_Iy_e, 0, 0]$  gegeben.

**Tabelle 6.3** Physikalische Parameter der Balldynamik.

Symbol	Physikalischer Parameter	Wert
$m_b$	Masse des Golfballs	0,046 kg
$g$	Gravitationskonstante	9,81 m/s <sup>2</sup>
$\mu_b$	Rollreibungskoeffizient des Balls auf dem Rasen	0,15

### Bestimmung eines optimalen Schlags

Ein Schlaggeschwindigkeitsvektor  $[{}_I\dot{x}_{B,0}, {}_I\dot{y}_{B,0}]^\top$  ist bezüglich einer Startposition des Balls  $[{}_Ix_{B,0}, {}_Iy_{B,0}]^\top$  genau dann optimal, wenn der Ball so geschlagen wird, dass er im Loch  $[{}_Ix_H, {}_Iy_H]^\top$  liegen bleibt, d. h.

$$\begin{bmatrix} {}_Ix_{B,e} \\ {}_Iy_{B,e} \end{bmatrix} = \begin{bmatrix} {}_Ix_H \\ {}_Iy_H \end{bmatrix}, \quad \begin{bmatrix} {}_I\dot{x}_{B,e} \\ {}_I\dot{y}_{B,e} \end{bmatrix} = \begin{bmatrix} {}_I\dot{x}_H \\ {}_I\dot{y}_H \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}. \quad (6.26)$$

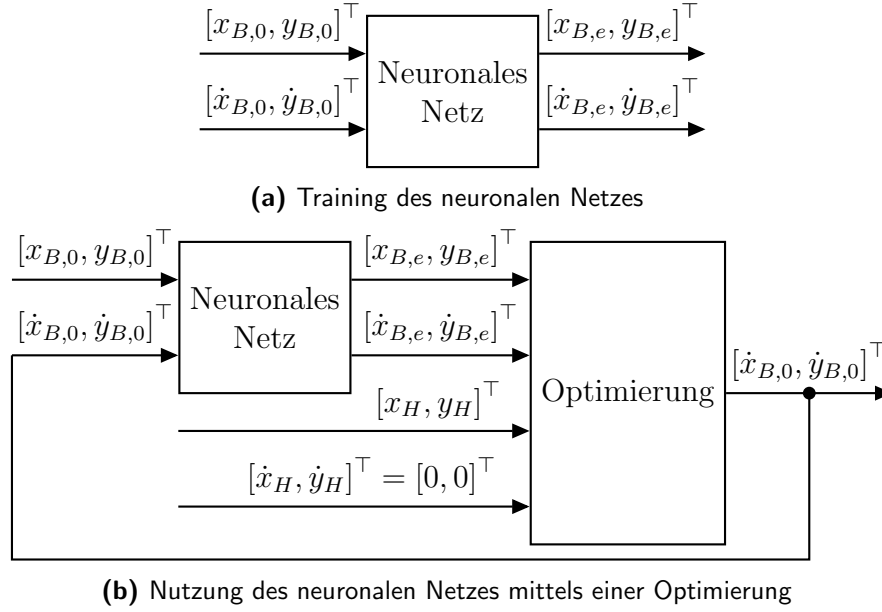
Die Bestimmung dieses optimalen Schlaggeschwindigkeitsvektors basiert auf einem neuronalen Netz. Der erste Ansatz verwendet ein flaches neuronales Netz aus 2 Schichten mit jeweils 30 versteckten Neuronen, das die Rolldynamik des Balls prädiziert, vgl. Abbildung 6.10(a), und anschließend die erforderliche Schlaggeschwindigkeit durch eine überlagerte Optimierung berechnet, vgl. Abbildung 6.10(b). Die

<sup>6</sup>Die numerische Integration erfolgt mittels des RK4-Solvers.

Optimierungsfunktion ist hierbei gegeben durch

$$J_b({}_I\dot{x}_{B,0}, {}_I\dot{y}_{B,0}) = ({}_I\mathbf{q}_{B,e} - {}_I\mathbf{q}_H)^\top \mathbf{W} ({}_I\mathbf{q}_{B,e} - {}_I\mathbf{q}_H) \quad (6.27)$$

mit  ${}_I\mathbf{q}_H = [{}_Ix_H, {}Iy_H, 0, 0]^\top$ ,  $\mathbf{W} = \text{diag}(1, 1, 1, 1)$ . Der Ausgang aus dem neuronalen Netz ist  ${}_I\mathbf{q}_{B,e}$ . Das Optimierungsproblem wird mittels einer Partikelschwarmoptimierung in Matlab gelöst.

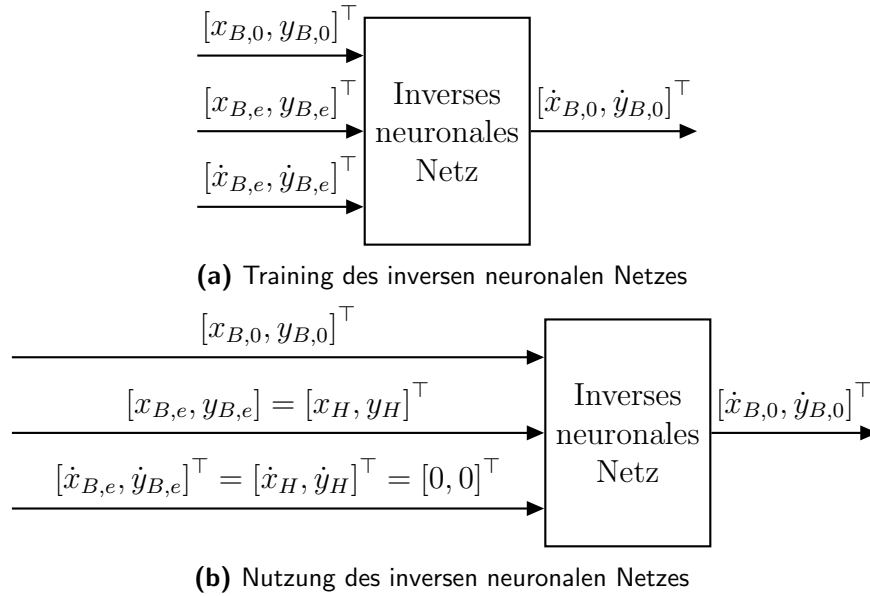


**Abbildung 6.10** Der erste Ansatz nutzt ein neuronales Netz zur Vorwärtsprädiktion der Roll-dynamik des Balls und ermittelt die Schlaggeschwindigkeit mittels einer überlagerten Optimierung. Alle Größen werden bezüglich des Inertialkoordinatensystems I angegeben.

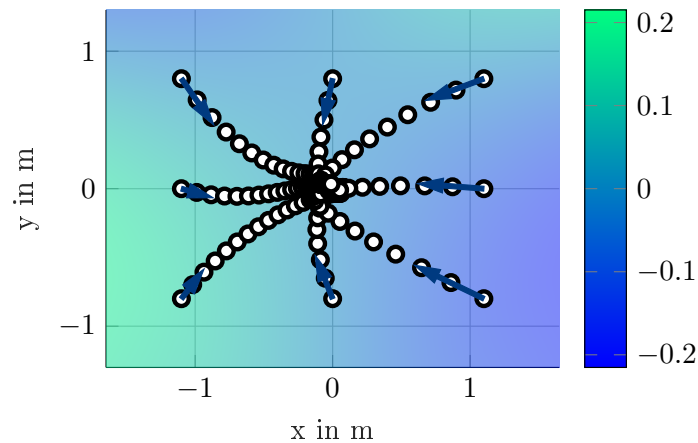
Ein alternativer neuartiger Ansatz besteht darin, die Netzwerkarchitektur aus der Abbildung 6.10(a) durch geschicktes Vertauschen der Ein- und Ausgänge so zu verändern, vgl. Abbildung 6.11(a), dass ein in Bezug auf die Schlagaufgabe *inverses* neuronales Netz entsteht, das unmittelbar der optimale Schlaggeschwindigkeitsvektor berechnet, vgl. Abbildung 6.11(b). Dadurch wird die aufwändige Lösung des Optimierungsproblems (6.27) obsolet. Dies führt zu einer verkürzten Rechenzeit bei einer vergleichbar hohen Vorhersagegenauigkeit wie beim ersten Ansatz. Die Abbildung 6.12 zeigt exemplarisch, dass die mittels des inversen neuronalen Netzes bestimmten Schlaggeschwindigkeiten simulativ in der Lage sind, den Ball so zu schlagen, dass er tatsächlich von unterschiedlichen Startpositionen aus zuverlässig in das Loch rollt. Dieses Ergebnis bestätigt die Machbarkeit des vorgestellten Ansatzes.

### 6.1.4 Ergebnisse

Im Folgenden wird der Ablauf eines autonomen Golfspiels beschrieben und der Erfolg der beschriebenen Methode evaluiert. Dies sind die Ergebnisse einer studentischen Arbeit, in der die Funktionsweise des Golfroboters auf Szenarien mit Hügeln erweitert wurde und diese auch ausführlich getestet worden sind [238].

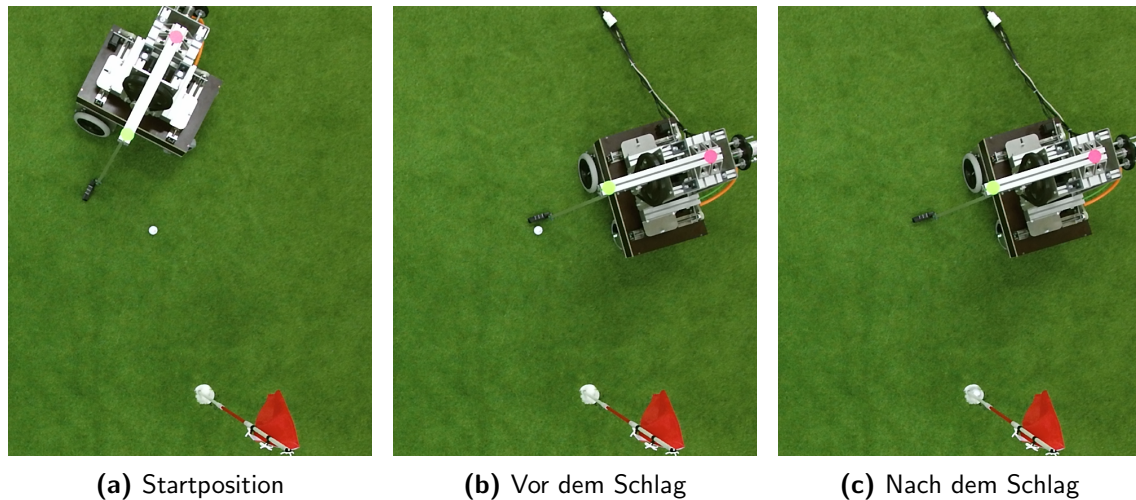


**Abbildung 6.11** Der zweite Ansatz nutzt ein inverses neuronales Netz, das unmittelbar die Schlaggeschwindigkeit für eine gegebene Spielsituation ermittelt. Alle Größen werden bezüglich des Inertialkoordinatensystems I angegeben.



**Abbildung 6.12** Simulativ bestimmte Balltrajektorien (weiße Kreise) für optimale Schlaggeschwindigkeiten (Pfeile in dunkelblau) für unterschiedliche Startpositionen des Balls unter Nutzung des inversen neuronalen Netzes. Das Loch wird im Ursprung des Koordinatensystems I angenommen und das Höhenprofil ist durch die Skala definiert.

Zunächst werden sämtliche Spielobjekte von der Spielfläche entfernt und ein Tiefenbild zur Höhenprofilbestimmung aufgenommen und als Spline-Funktion approximiert. Auf dieser Basis werden zufällige Trainingsschläge simulativ erzeugt und für das Training des neuronalen Netzes genutzt. Anschließend werden die Spielobjekte (Golfs, Ball) beliebig auf der Spielfläche platziert. Diese Ausgangssituation ist in der Abbildung 6.13(a) dargestellt. Von hier ausgehend werden die Algorithmen für das autonome Golfspiel (Objektdetektion, Bestimmung des optimalen Schlags, Berechnung der Fahrstrategie) in Matlab ausgeführt und Zwischenergebnisse visualisiert. Nachdem der Golfroboter sich geeignet neben dem Ball platziert hat, vgl. Abbildung 6.13(b), wird der Schlag ausgeführt, vgl. Abbildung 6.13(c).



**Abbildung 6.13** Beispielhafte Situation und Ablauf eines Golfspiels des Roboters.

Dieses Beispielszenario demonstriert qualitativ das erfolgreiche Putten ohne hügelige Spielfläche. Um das Können des Roboters quantitativ zu bestimmen, wurden mehrere Versuchsreihen aus unterschiedlich schwierigen Spielsituationen durchgeführt [238]. Als einfachste Schwierigkeitsstufe wurde das Putten ohne notwendige Positionierung und ohne Berücksichtigung einer hügeligen Spielfläche festgelegt. In der zweiten Schwierigkeitsstufen wird der Roboter nicht perfekt ausgerichtet positioniert, sodass ein Fahrmanöver notwendig ist. Im dritten Fall beeinflusst eine hügelige Spielfläche die Berechnung des Schlagvektors (vgl. Unterabschnitt 6.1.3) und erst in der vierten Schwierigkeitsstufe wird zusätzlich angenommen, dass beim Fahren besonders hohe Hügel umfahren werden müssen.

Im Rahmen der Versuchsreihen wurden verschiedene Spielsituationen erstellt, die in diese vier Schwierigkeitsstufen eingeteilt werden können. Die Grafik Abbildung 6.13 zeigt bspw. eine Situation der Schwierigkeitsstufe 2. Um die Schwierigkeit detaillierter zu beschreiben, wurden jeder Schwierigkeitsstufe eine Punktzahl zugewiesen, die auf der Anzahl der benötigten Fähigkeiten basiert, die der Roboter für das Lösen einer Spielsituation benötigt. Ein Beispiel für eine solche Fähigkeit ist das Erkennen des Höhenprofils, welches für Spielsituationen der Schwierigkeitsstufen 1 und 2 nicht notwendig ist. Zusätzlich zur Punktzahl der benötigten Fähigkeiten geht die Distanz zwischen Ball und Loch als Faktor in den finalen Schwierigkeitswert ein. Aufgrund inhomogener Oberflächenbeschaffenheit wird angenommen, dass es schwieriger ist einen Ball über eine größere Distanz präzise zu putten, wie es auch bei einem realen Golfspiel der Fall ist. Diese Annahmen bilden die Grundlage eines situationsabhängigen Schwierigkeitswerts.

Für die Durchführung der Versuchsreihen in denen jeweils eine Spielsituation zehn Mal gelöst wurde, wurden Spielregeln festgelegt. Trifft der Golfroboter beim ersten Versuch das Loch nicht, darf die neu entstandene Spielsituation gelöst werden, sofern sie lösbar ist. Es entstehen Erfolgsquoten für den ersten und für den zweiten Versuch. Ein dritter Versuch wird nicht durchgeführt. Wie auch im realen Golfspiel werden mehr benötigte Versuche schlechter bewertet, weshalb der zweite Versuch beim Mitteln beider Erfolgsquoten niedriger gewichtet wird. Mit dieser Auswertungsregel und quantifizierbaren Spielsituationen wurde das Können des Roboters differenziert er-



mittelt. Anhand von sechs Versuchsreihen konnte gezeigt werden, dass Situationen, die als schwerer eingestuft werden, auch weniger erfolgreich gelöst werden können. Zwischen Situationen mit knapp 100%iger Erfolgsquote, bis zu Spielsituationen bei denen immer zwei Versuche benötigt wurden, konnte ein nahezu linearer Zusammenhang festgestellt werden. Dies spricht dafür, dass die beschriebene Auswertungsmethodik auch in Zukunft für die Validierung des Roboters genutzt werden kann und potentiell auch bei anderen Golfrobotern anwendbar ist.

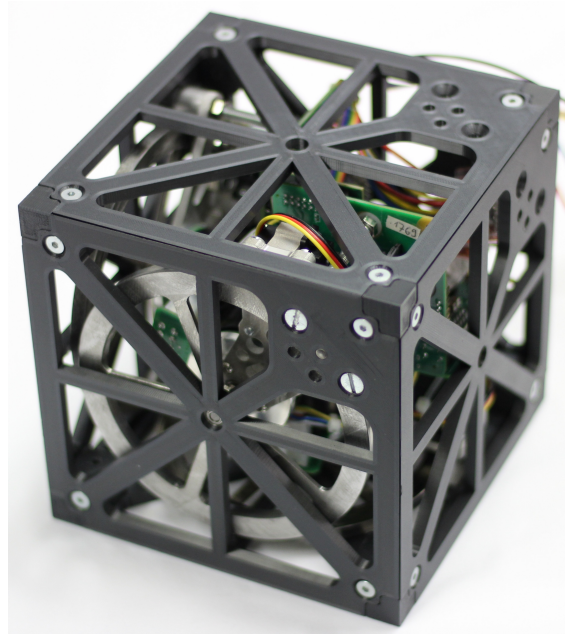
Spielsituationen der Schwierigkeitsstufe 3 erfordern ähnliches Können wie beim menschlichen Putten. Mit einer relativ einfachen Positionierung ohne Hügel umfahren zu müssen, diese jedoch beim Schlag zu berücksichtigen, sind diese Situationen vergleichbar mit den Fähigkeiten, die ein Mensch benötigt um solche Spielsituationen zu lösen. Für eine Distanz von 1,5 m zwischen Ball und Loch konnte mit nur einem Versuch eine Erfolgsquote von 60 % erreicht werden, was vergleichbar ist mit fortgeschrittenen Golfspieler\*innen [239].

## 6.2 Selbstbalancierender Würfel

In diesem Abschnitt wird der Entwurf und Aufbau eines Demonstrators für das Projekt DART beschrieben. Es handelt sich um einen selbstbalancierenden Würfel, der in der Projektlaufzeit als regelungstechnisches Testsystem entwickelt wurde, siehe auch Abbildung 6.14. Ebenso wie der Golfroboter dient er als reales System zur Testung der in DART entwickelten Methoden. In Unterabschnitt 6.2.1 werden der mechatronische Aufbau, die Manöver und die Entwicklung des Würfels beschrieben. Danach folgt der modellbasierte Regelungsentwurf in Unterabschnitt 6.2.2. Zum Schluss werden in Abschnitt 6.2.2 die Ergebnisse der klassischen Regelung mittels LQ-Regler und einige der Ergebnisse der neu entwickelten hybriden Methoden dargestellt.

### 6.2.1 Mechatronischer Entwurf des Würfels

Das Regelungsziel des balancierenden Würfels ist die automatisierte Stabilisierung auf einer seiner Kanten oder Ecken. Darauf ist der mechatronische Entwurf mit Sensorik, Aktorik und Informationsverarbeitung ausgelegt, sodass er aus einer beliebigen Orientierung von allein auf eine seiner Kanten oder Ecken aufschwingen kann und sich dort stabilisiert. Als Aktoren werden Schwungräder verwendet, ein bekannter Aktor in mechanischen System. Ein Anwendungsbeispiel sind Satelliten, die durch die Geschwindigkeitsänderung von Schwungrädern ein Drehmoment erfahren und somit ihre Orientierung ändern können. Als kleines, einfach zu handhabendes und aus Standardteilen entwickeltes System kann der Würfel von anderen nachgebaut werden und einfach in der Lehre und Forschung verwendet werden. Ursprünglich wurde er u. a. von dem Institut "Institute for Dynamic Systems and Control" der ETH Zürich entwickelt [240]. Der Würfel des DART-Projekts wurde selbst konstruiert und entwickelt, sodass sich der mechatronische Aufbau unterscheidet, jedoch funktioniert er nach dem selben Prinzip.



**Abbildung 6.14** Der selbstbalancierende Würfel als mechatronisches System zum Testen der entwickelten hybriden Methoden. Der Würfel hat eine Kantenlänge von 15 cm.

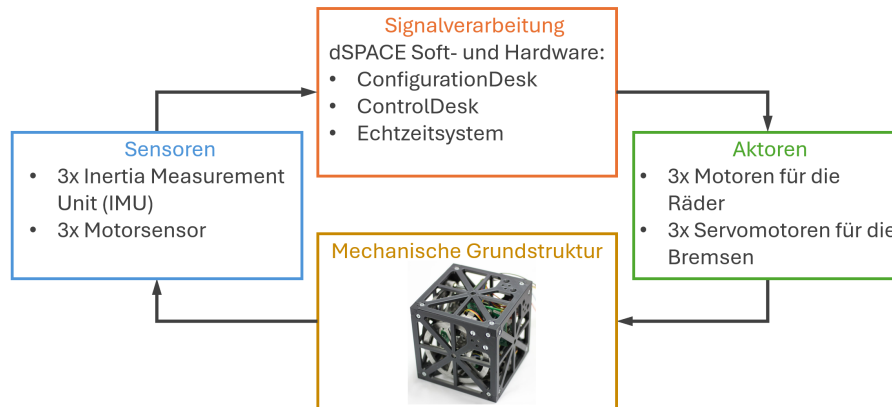
### Mechatronischer Aufbau

In Abbildung 6.15 wird der mechatronische Aufbau des Würfels veranschaulicht. Er beinhaltet die mechanische Grundstruktur, Sensoren, Informationsverarbeitung und Aktoren. Die mechanische Grundstruktur besteht im Wesentlichen aus dem Rahmen und den inneren Verbindungselementen. Der Rahmen des Würfels wird durch sechs additiv gefertigte Kunststoffplatten gebildet, die an den Ecken miteinander verschraubt sind. Die restlichen Komponenten sind an dem Rahmen im Inneren des Würfels angebracht. Ein Kabelstrang verläuft aus dem Würfel zur Verbindung zum Echtzeitsystem und zur Stromversorgung. Dabei muss beachtet werden, dass der Kabelstrang nicht die Bewegung des Würfels beeinflusst. Ein zukünftiges Ziel könnte sein, den Würfel vollständig autonom zu gestalten, so dass die Stromversorgung etc. im Inneren integriert werden müsste. Die wesentlichen mechanischen Komponenten sind die drei Schwungräder (im Folgenden als Räder bezeichnet) und die drei Scheibenbremsen. Jedes Rad wird durch einen Elektromotor angetrieben und kann durch eine Scheibenbremse abrupt gebremst werden, die durch jeweils einen Servomotor aktiviert wird.

Für die Erkennung der Bewegung des Würfels werden als Sensoren drei IMU<sup>7</sup> und drei kontaktlose, magnetische Sensoren<sup>8</sup> für die Elektromotoren, im Folgenden als Motorsensoren bezeichnet. Der IMU misst die drei translatorischen Beschleunigungen und drei rotatorischen Geschwindigkeiten. Durch ein Filter können aus den gemessenen Signalen dieser Sensoren die drei Winkel  $\theta_x$ ,  $\theta_y$  und  $\theta_z$  des Würfels im Raum bezogen auf das Inertialsystem bestimmt werden, siehe Abbildung 6.18 für eine Visualisierung der Winkel. Der Motorsensor liefert die Position  $\psi_i$  und Geschwindigkeit  $\dot{\psi}_i$  eines Rades. Zur Informationsverarbeitung wird Matlab/Simulink

<sup>7</sup>Adafruit 2472 BNO055 Inertia Measurement Unit

<sup>8</sup>AS5045B von ams OSRAM

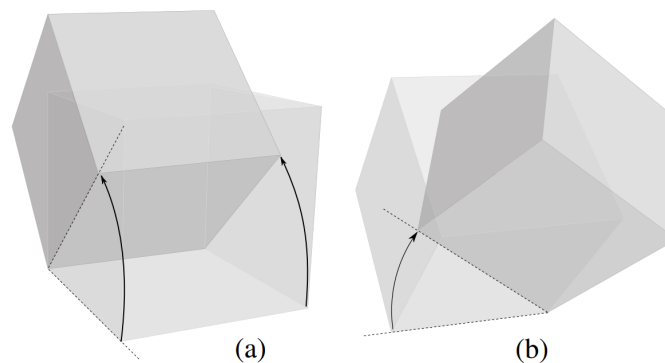


**Abbildung 6.15** Struktur des Würfels als mechatronisches System

und ein Scalexio Echtzeitsystem mit entsprechender Software von dSPACE verwendet. In Matlab/Simulink ist das Modell des Systems und der Regelkreis implementiert. Die Signalübertragung zwischen dem Modell und dem Würfel wird über das Echtzeitsystem und die Software ConfigurationDesk und ControlDesk realisiert. Die Aktoren sind drei bürstenlose 80-Watt Elektromotoren<sup>9</sup> und drei Servomotoren<sup>10</sup>.

### Beschreibung des Manövers

Das betrachtete Manöver des Würfels beginnt auf einer seiner Flächen, es folgt der Aufschwung und die Stabilisierung auf eine Kante und endet mit dem Aufschwung und der Stabilisierung auf eine Ecke, was in Abbildung 6.16 visualisiert ist. Die Regelung wird aktiv, sobald sich der Würfel durch den Aufschwung im Einzugsbereich der oberen Ruhelage befindet.

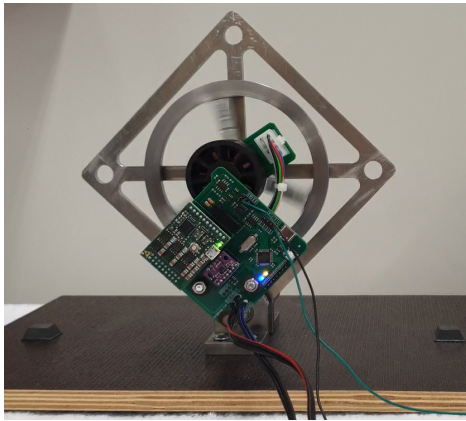


**Abbildung 6.16** Veranschaulichung des Manövers: a) von der Fläche auf die Kante und b) von der Kante auf die Ecke. Die Abbildung wurde aus [241] übernommen.

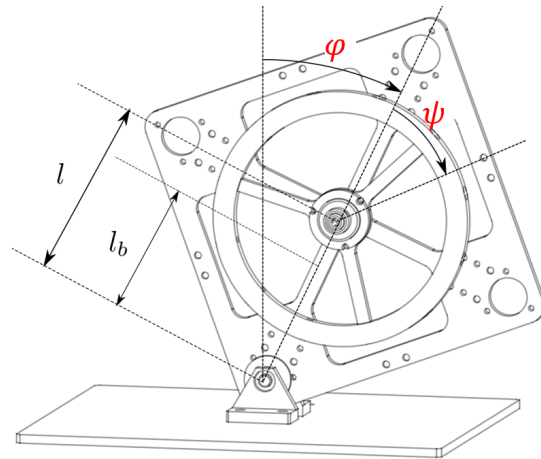
Beim Aufschwung werden je nach Manöver ein oder mehrere Schwungräder auf eine bestimmte Drehgeschwindigkeit beschleunigt, damit ausreichend kinetische Energie vorhanden ist. Für den Aufschwung von der Fläche auf die Kante ist nur ein Rad notwendig, das sich um die Achse der Kante dreht, über die aufgeschwungen wird. Für den Aufschwung von der Kante auf die Ecke sind zusätzlich die anderen

<sup>9</sup>45-flat der Maxon Motor AG

<sup>10</sup>Hitec Servo D-89 MW



(a) Der Prototyp während des Balancierens.



(b) Kinematische Größen des Prototypens. Abbildung aus [241]

**Abbildung 6.17** Der Prototyp als Vorstufe des Würfels zum Testen des mechatronischen Aufbaus.

zwei Räder notwendig. Die Räder werden durch die Scheibenbremse abrupt bis zum Stillstand abgebremst, sodass die kinetische Energie der Räder aufgrund der Energieerhaltung auf den Rahmen des Würfels in Form eines Momentes übertragen wird. Der Würfel erfährt eine rotatorische Beschleunigung und schwingt über eine Kante oder Ecke hoch in den Einzugsbereich des Reglers der instabilen Ruhelage. Experimente haben gezeigt, dass der Einzugsbereich der Regelung innerhalb von circa  $15^\circ$  um die Ruhelage liegt. Durch die Regelung wird das Rad durch den Elektromotor in die passende Richtung beschleunigt, sodass es dem Umfallen des Würfels entgegenwirkt und ihn wieder in die instabile Ruhelage zurückführt. Aufgrund des dritten Newtonschen Axioms, auch das Prinzip von Aktion und Reaktion genannt, wirkt ein entgegengesetztes, gleichgroßes Moment auf den Würfel, wenn sich das Rad in eine Richtung dreht. Der Würfel erfährt durch das Rad eine Beschleunigung und wird zurück in die Richtung der instabilen Ruhelage bewegt.

### Entwicklung des Demonstrators

Um das grundsätzliche Wirkprinzip des Demonstrators kennenzulernen, wurde im Rahmen des Projekt zunächst ein vereinfachter Prototyp aufgebaut. Der Prototyp in Abbildung 6.17(a) besteht nur aus einer Fläche des gesamten Würfels, mit einem Rad, einer Bremse und einem IMU. Er ist an einer Ecke der Fläche drehbar gelagert und hat einen rotatorischen Freiheitsgrad. Mit dem Prototyp kann der Aufschwung und die Stabilisierung des Würfels auf der Kante getestet werden. Er erlaubt das Testen des mechatronischen Aufbaus in einer geringeren Komplexität als der gesamte Würfel und die Ergebnisse sind auf den Würfel übertragbar.

Als nächstes wird auf einige Aspekte bei der Konstruktion des Würfels eingegangen, die zu beachten sind, damit der Aufschwung und die Stabilisierung möglich werden bzw. besser funktionieren. Um eine möglichst geringe Masse bewegen zu müssen, wurde der Rahmen des Würfels aus Kunststoff additiv gefertigt. Die inneren, selbst konstruierten Teile sind aus Aluminium. Die Schwungräder sind aus

Stahl, um möglichst viel Drehmoment für den Aufschwung zu erhalten. Für den Aufschwung ist ein kleines Massenträgheitsmoment des Würfels vorteilhaft, weil dann weniger Energie benötigt wird. Deswegen sollte der Würfel so konstruiert werden, dass die gesamte Masse gering ist, die Länge der Hebel möglichst klein ist und der Massenmittelpunkt nah an den Drehachsen liegt. Dabei muss beachtet werden, dass für alle Komponenten im Inneren des Würfels genug Raum ist.

Bei der Konstruktion des Rades gibt es gegensätzliche Anforderungen. Zum einen wird für den Aufschwung des Würfels Energie benötigt, die das Rad liefern soll. Dafür ist ein Rad mit ausreichend großem Massenträgheitsmoment notwendig, so dass das Rad genug kinetische Energie für den Aufschwung des Würfels liefert. Bei der Stabilisierung des Würfels muss zum anderen das Rad schnell beschleunigt werden. Dafür ist ein Rad mit geringem Massenträgheitsmoment vorteilhaft. Um diesen Zielkonflikt zu lösen, muss ein Rad mit passendem Massenträgheitsmoment gefunden werden.

### 6.2.2 Modellbasierter Regelungsentwurf des Würfels

Die Modellbildung des Würfels wird mit Hilfe der Bewegungsgleichungen aus dem Prinzip von Jourdain hergeleitet. Es wird die allgemeine Gleichung eines mechanischen Mehrkörpersystems

$$\mathbf{M}\ddot{\mathbf{q}} + \mathbf{C}\dot{\mathbf{q}} + \mathbf{G} = \begin{bmatrix} \mathbf{0}_{3 \times 3} \\ \mathbf{I}_{3 \times 3} \end{bmatrix} \mathbf{u}, \quad (6.28)$$

mit den generalisierten Koordinaten

$$\mathbf{q} = [\theta_1, \theta_2, \theta_3, \psi_x, \psi_y, \psi_z]^T \in \mathbb{R}^{6 \times 1} \quad (6.29)$$

verwendet. Dabei beziehen sich die drei räumlichen Winkel  $\theta_i$  auf das Inertialsystem des Würfelkörpers und die Winkel  $\psi_i$  auf das jeweilige Rad, das sich um die  $W_x e_x$ -,  $W_y e_y$ -, oder  $W_z e_z$ -Achse dreht. Die zeitlichen Ableitungen der generalisierten Koordinaten ergibt sich als

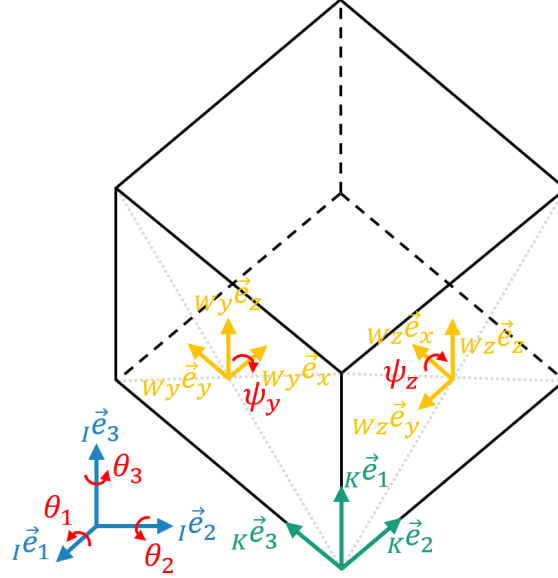
$$\dot{\mathbf{q}} = [\dot{\theta}_1, \dot{\theta}_2, \dot{\theta}_3, \dot{\psi}_x, \dot{\psi}_y, \dot{\psi}_z]^T \in \mathbb{R}^{6 \times 1} \quad (6.30)$$

und die Eingänge des Systems

$$\mathbf{u} = [u_1, u_2, u_3]^T \in \mathbb{R}^{3 \times 1} \quad (6.31)$$

sind die Ströme  $u_i$  an dem jeweiligen Motor des  $i$ -ten Rades,  $i = 1, 2, 3$ . Das Motordrehmoment  $T_{m,i} = K_m u_i$  des  $i$ -ten Rades berechnet sich aus der Motorkonstante  $K_m$  und der Eingangsspannung  $u_i$ .  $\mathbf{0}_{3 \times 3}$  und  $\mathbf{I}_{3 \times 3}$  stellen die Null- bzw. Einheitsmatrix dar. Die Massenmatrix

$$\mathbf{M} = \sum_{i=1}^{n_k} (m_i \mathbf{J}_{Ti}^T \mathbf{J}_{Ti} + \mathbf{J}_{Ri}^T \boldsymbol{\Theta}_i \mathbf{J}_{Ri}) \in \mathbb{R}^{6 \times 6} \quad (6.32)$$



**Abbildung 6.18** Schematische Darstellung des Würfels auf der Ecke mit den Koordinatensystemen: Inertialsystem  $I$ , körperfestes System  $K$  des Würfels und körperfeste Systeme  $W_i$  des  $x$ -,  $y$ - und  $z$ -Rades. Der Übersichtlichkeit halber ist das  $W_x$ -System in der  ${}_{K}e_2$ - ${}_{K}e_3$ -Ebene nicht dargestellt.

setzt sich unter anderem zusammen aus der Jakobimatrix der Translation

$$\mathbf{J}_{Ti} = \frac{\partial \dot{\mathbf{r}}_{0i}}{\partial \dot{\mathbf{q}}^T} \in \mathbb{R}^{3 \times 6}, \quad (6.33)$$

der Jakobimatrix der Rotation

$$\mathbf{J}_{Ri} = \frac{\partial \boldsymbol{\omega}_i}{\partial \dot{\mathbf{q}}^T} \in \mathbb{R}^{3 \times 6}, \quad (6.34)$$

der Geschwindigkeit  $\dot{\mathbf{r}}$  des Schwerpunkts der Masse  $m_i$ , der Winkelgeschwindigkeit  $\boldsymbol{\omega}_i$  und des Massenträgheitsmoments  $\boldsymbol{\Theta}_i \in \mathbb{R}^{3 \times 3}$  des  $i$ -ten Körpers um seinen Schwerpunkt. Der Index  $i = 1, \dots, n_k$  läuft über alle Körper der Anzahl  $n_k = 4$  (Würfel und 3 Scheiben). In dieser Darstellung sind die translatorischen Größen bezüglich des Inertialsystems und die rotatorischen Größen bezüglich des körperfesten Systems angegeben. Die Matrix  $\mathbf{C} = \sum_{i=1}^{n_q} \mathbf{C}_i(\mathbf{q}, \dot{\mathbf{q}}) \in \mathbb{R}^{6 \times 6}$  beinhaltet die Zentrifugal- und Corioliskräfte, wobei  $n_q = 6$  die Anzahl der generalisierten Koordinaten  $\mathbf{q}$  ist. Die Elemente der Matrizen  $\mathbf{C}_i(\mathbf{q}, \dot{\mathbf{q}})$  der einzelnen Körper werden beschrieben durch

$$\mathbf{c}_{i,nm} = \sum_{k=1}^{n_q} \mathbf{c}_{i,nmk} \dot{\mathbf{q}}_k \quad \text{für} \quad \begin{array}{ll} n &= 1, \dots, 6 \\ m &= 1, \dots, 6 \end{array} \quad (6.35)$$

mit dem sogenannten Christoffelsymbol der ersten Art

$$\mathbf{c}_{i,nmk} = \frac{1}{2} \left( \frac{\partial \mathbf{M}_{i,mn}}{\partial \mathbf{q}_k} + \frac{\partial \mathbf{M}_{i,mk}}{\partial \mathbf{q}_n} - \frac{\partial \mathbf{M}_{i,nk}}{\partial \mathbf{q}_m} \right). \quad (6.36)$$

Die Matrix

$$\mathbf{G} = \left( -\frac{\partial U(\mathbf{q})}{\partial \mathbf{q}} \right)^T \in \mathbb{R}^{6 \times 1} \quad (6.37)$$

bringt die Gravitationskräfte in die Bewegungsgleichung ein. Dabei ist

$$U = \mathbf{g}_K^T \sum_{i=1}^{n_k} m_i \mathbf{r}_{0i} \quad (6.38)$$

die potenzielle Energie des Systems. Sie setzt sich zusammen aus einer Summe der Massen  $m_i$  und deren Ortsvektoren  $\mathbf{r}_{0i}$  im Inertialsystem und dem Gravitationsvektor

$$\mathbf{g}_K = \mathbf{R}_{K0} \mathbf{g}_0 \quad (6.39)$$

im körperfesten Koordinatensystem.  $\mathbf{g}_K$  wird durch eine Koordinatentransformation des Gravitationsvektors

$$\mathbf{g}_0 = [0, 0, -g]^T \quad (6.40)$$

im Inertialsystem durch die Rotationsmatrix

$$\begin{aligned} \mathbf{R}_{K0} &= \mathbf{R}_x(\alpha) \mathbf{R}_y(\beta) \mathbf{R}_z(\gamma) \\ &= \begin{bmatrix} 1 & 0 & 0 \\ 0 & c(\alpha) & -s(\alpha) \\ 0 & s(\alpha) & c(\alpha) \end{bmatrix} \begin{bmatrix} c(\beta) & 0 & s(\beta) \\ 0 & 1 & 0 \\ -s(\beta) & 0 & c(\beta) \end{bmatrix} \begin{bmatrix} c(\gamma) & -s(\gamma) & 0 \\ s(\gamma) & c(\gamma) & 0 \\ 0 & 0 & 1 \end{bmatrix} \\ &= \begin{bmatrix} c(\beta)c(\gamma) & c(\alpha)s(\gamma) + c(\gamma)s(\alpha)s(\beta) & s(\alpha)s(\gamma) - c(\alpha)c(\gamma)s(\beta) \\ -c(\beta)s(\gamma) & c(\alpha)c(\gamma) - s(\alpha)s(\beta)s(\gamma) & c(\gamma)s(\alpha) + c(\alpha)s(\beta)s(\gamma) \\ s(\beta) & -c(\beta)s(\alpha) & c(\alpha)c(\beta) \end{bmatrix} \end{aligned} \quad (6.41)$$

der  $xyz$ -Kardanwinkel erhalten. Die Rotationsmatrix  $\mathbf{R}_{K0}$  setzt sich zusammen aus den drei aufeinanderfolgenden Rotationen um die  $x$ -,  $y$ - und  $z$ -Achse. In der Rotationsmatrix  $\mathbf{R}_{K0}$  ist  $s(\cdot) := \sin(\cdot)$  und  $c(\cdot) := \cos(\cdot)$ . Für die Reglerauslegung in einer instabilen Ruhelage wird die nichtlineare Dynamikgleichung (6.28) in eine lineare Zustandsraumbeschreibung

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u} \quad (6.42)$$

überführt, wobei der Zustandsvektor

$$\mathbf{x} = [\theta_1, \theta_2, \theta_3, \dot{\theta}_1, \dot{\theta}_2, \dot{\theta}_3, \psi_x, \psi_y, \psi_z]^T \quad (6.43)$$

eingeführt wird. Es ergibt sich die Gesamtsystembeschreibung

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \mathbf{u}) = \mathbf{f}(\mathbf{x}) + \mathbf{g}(\mathbf{x})\mathbf{u} \quad (6.44)$$

$$= \begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \\ \dot{\theta}_3 \\ \mathbf{M}^{-1}(-\mathbf{G} - \mathbf{C}\dot{\mathbf{q}}) \end{bmatrix} + \begin{bmatrix} \mathbf{0}_{3 \times 3} \\ \mathbf{0}_{3 \times 3} \\ \mathbf{I}_{3 \times 3} \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \\ u_3 \end{bmatrix}. \quad (6.45)$$

Die Systemmatrix  $\mathbf{A}$  und die Eingangsmatrix  $\mathbf{B}$  der Zustandsdifferentialgleichungen werden durch einen Linearisierungsansatz erhalten

$$\mathbf{A} = \frac{\partial \mathbf{f}(\mathbf{x}, \mathbf{u})}{\partial \mathbf{x}}, \quad (6.46)$$

$$\mathbf{B} = \frac{\partial \mathbf{f}(\mathbf{x}, \mathbf{u})}{\partial \mathbf{u}}. \quad (6.47)$$

Mit dem linearisierten System wird ein LQ-Regler [174] entworfen. Die Zielfunktion ist

$$J = (\mathbf{x}_g - \mathbf{x}(t_e))^T \mathbf{S} (\mathbf{x}_g - \mathbf{x}(t_e)) + \int_0^{t_e} \mathbf{x}^T \mathbf{Q} \mathbf{x} + \mathbf{u}^T \mathbf{R} \mathbf{u} \, dt, \quad (6.48)$$

mit dem Zielzustand  $\mathbf{x}_g$ , der den Zustand des Systems auf der Kante oder Ecke beschreibt, der Endzeit  $t_e$  und den Gewichtungsmatrizen  $\mathbf{S}$ ,  $\mathbf{Q}$ , und  $\mathbf{R}$ .

## Ergebnisse

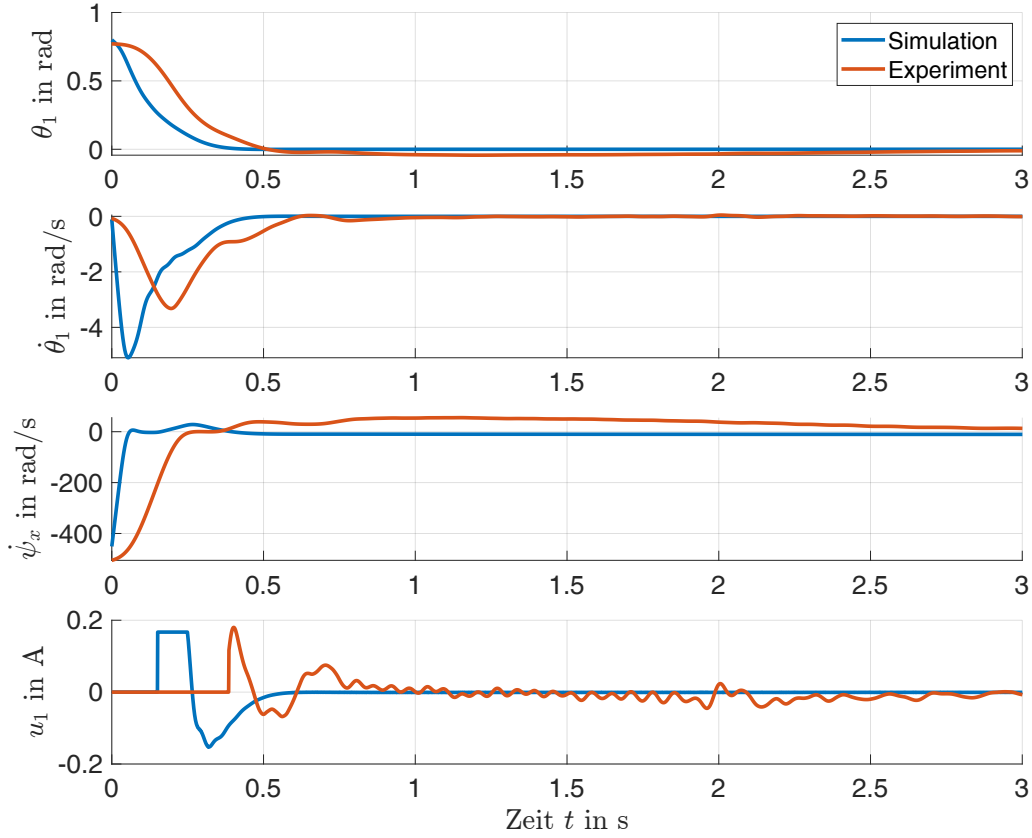
Nach der Systembeschreibung des selbstbalancierenden Würfels werden in diesem Abschnitt Ergebnisse verschiedener Experimente vorgestellt. Zunächst wird der Würfel mittels klassischer Entwurfsverfahren der Regelungstechnik in Betrieb genommen und anschließend die PCHD-Modelle aus Abschnitt 3.4 und die Optimalsteuerung aus Unterabschnitt 4.4.1 als hybride Methoden getestet.

### Klassische Entwurfsverfahren

Beim klassischen Entwurf mittels LQ-Regelung wird der Aufschwung und die Stabilisierung des Würfels von der Fläche auf die Kante untersucht und die Ergebnisse sind in Abbildung 6.19 zu sehen. Für dieses Manöver sind die relevanten Zustände der Würfelwinkel  $\theta_1$ , dessen Winkelgeschwindigkeit  $\dot{\theta}_1$ , die Winkelgeschwindigkeit  $\dot{\psi}_x$  des Rades und der Eingang  $u_1$  als Motorstrom. Der Würfel beginnt auf der Fläche bei  $\theta_1 = 45^\circ$  und schwingt über die Kante in die obere instabile Ruhelage bei  $\theta_1 = 0^\circ$ .

Der Bremsvorgang mit dem abrupten Abbremsen des Schwungrades wird von uns in der Modellierung als ideal angenommen. Da ein sofortiges Abbremsen in der Realität nicht umsetzbar ist, wird eine vollständige Übereinstimmung von Modell und realem Verlauf des Experiments nicht erwartet. Der Nachweis der Funktionsfähigkeit des Würfels wird daher durch den funktionierenden Aufschwung und die Stabilisierung beurteilt. In Abbildung 6.19 wird deutlich, dass in der Simulation der Aufschwung schneller als im Experiment ist. Das liegt an der getroffenen Annahme des idealen Bremsens in der Modellierung des komplexen Bremsvorgangs. Außerdem ist in der Simulation eine Stellgrößenbeschränkung des Eingangs  $u_1$  von 0,17 A implementiert, die ungefähr dem realen Verhalten der Bremse entspricht. Durch den langsameren Aufschwung im Experiment beginnt die Regelung später, was am Verlauf von  $u_1$  erkennbar ist. Die Regelung oszilliert geringfügig, was bei einem realen Stabilisierungsvorgang zu erwarten ist. Das Experiment zeigt, dass der Würfel mittels klassischer Entwurfsverfahren erfolgreich auf der Kante stabilisiert werden kann.





**Abbildung 6.19** Aufschwung und Stabilisierung des Würfels von der Fläche auf die Kante mit den klassischen Entwurfsverfahren. Die Regelung wird hier durch einen LQ-Regler umgesetzt.

### Hybride Methoden

Beispielhaft werden im Folgenden zwei Entwurfsschritte adressiert und durch hybride Methoden ersetzt. Dies sind einerseits die datengetriebene Formulierung des Würfelmodells als PCHD-Modell basierend auf den Methoden aus Abschnitt 3.4 und andererseits die Anwendung der hybriden Optimalsteuerung nach Unterabschnitt 4.4.1 auf den Würfel, um den Aufschwung auf die Kante zu realisieren.

**Datengetriebene Modellbildung mit PCHD-Modellen** Zur vereinfachten Veranschaulichung soll die Bewegung von der Kante auf die Ecke des eindimensionalen Prototypens betrachtet werden, dessen Bewegungsgleichungen durch

$$\ddot{\varphi} = \frac{(m_b l_b + m_w l)g \sin \varphi - T_m - C_b \dot{\varphi} + C_w \dot{\psi}}{\Theta_b + m_w l^2}, \quad (6.49)$$

$$\ddot{\psi} = \frac{(\Theta_b + \Theta_w + m_w l^2)(T_m - C_w \dot{\psi})}{\Theta_w(\Theta_b + m_w l^2)} - \frac{(m_b l_b + m_w l)g \sin \varphi - C_b \dot{\varphi}}{(\Theta_b + m_w l^2)} \quad (6.50)$$

beschrieben werden und dessen Parameter in der Tabelle 6.4 zu finden sind. Dabei berechnet sich das Motordrehmoment  $T_m = K_m u$  aus der Motorkonstanten  $K_m$  und dem Eingangsstrom  $u$ . Ausgehend von der Energiefunktion

$$V(\mathbf{x}) = \frac{1}{2}(\Theta_w + a)x_2^2 + \frac{1}{2}\Theta_w x_2 x_3 + b g \cos(x_1), \quad (6.51)$$

mit  $a = \Theta_b + m_w l^2$ ,  $b = m_b l_b + m_w l$  und dem Zustand  $\mathbf{x} = [\varphi, \dot{\varphi}, \psi]^T$  kann die Bibliothek  $\Psi$  nach Gleichung (3.75a) durch den Gradienten der Energiefunktion gebildet werden:

$$\Psi(\mathbf{x}) = \left( \frac{\partial V}{\partial \mathbf{x}} \right)^T = \begin{bmatrix} -bg \sin(x_1) \\ (\Theta_w + a)x_2 + \Theta_w x_3 \\ \Theta_w x_3 + \Theta_w x_2 \end{bmatrix}. \quad (6.52)$$

Das analytisch bestimmte PCHD-Modell kann nach Gleichung (3.75a) daraufhin durch die Matrizen  $\mathbf{J}$ ,  $\mathbf{D}$  und den Vektor  $\mathbf{b}$  definiert werden, die folgendermaßen lauten:

$$\mathbf{J} = \begin{bmatrix} 0 & \frac{1}{a} & \frac{1}{a} \\ \frac{1}{a} & 0 & 0 \\ \frac{1}{a} & 0 & 0 \end{bmatrix}, \mathbf{D} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & \frac{C_b + C_w}{a^2} & -\frac{\Theta_w + a}{\Theta_w a^2} C_w - \frac{C_b}{a^2} \\ 0 & -\frac{\Theta_w + a}{\Theta_w a^2} C_w - \frac{C_b}{a^2} & \frac{(\Theta_w + a)^2}{(\Theta_w a)^2} C_w + \frac{C_b}{a^2} \end{bmatrix}, \mathbf{b} = \begin{bmatrix} 0 \\ -\frac{1}{a} \\ \frac{\Theta_w + a}{\Theta_w a} \end{bmatrix}. \quad (6.53)$$

Das datengetriebene PCHD-Modell kann mithilfe gesammelter Messdaten des Würfels durch den Algorithmus, der in Abbildung 3.22 dargestellt ist und detailliert in Unterabschnitt 3.4.2 beschrieben wird, bestimmt werden. Dieses stimmt genau mit dem analytisch bestimmten PCHD-Modell überein (vgl. Gleichung (6.53)). Mithilfe dieses datengetriebenen PCHD-Modells lässt sich anschließend eine passivitätsbasierte Regelung entwerfen (vgl. Abschnitt 4.3).

Physikalischer Parameter	Symbol	Wert
Abstand Drehpunkt und Motorachse	$l$	0,089 45 m
Abstand Drehpunkt und Massenmittelpunkt	$l_b$	0,077 86 m
Masse des Körpers ohne Rad	$m_b$	0,573 kg
Masse des Rades	$m_w$	0,195 kg
Gravitationsbeschleunigung	$g$	9,81 m s <sup>-2</sup>
Motorkonstante	$K_m$	40,4 · 10 <sup>-3</sup> Nm A <sup>-1</sup>
Gleitreibungskoeffizient des Körpers	$C_b$	0,7 · 10 <sup>-3</sup> kg m <sup>2</sup> s <sup>-1</sup>
Gleitreibungskoeffizient des Rades	$C_w$	0,05 · 10 <sup>-3</sup> kg m <sup>2</sup> s <sup>-1</sup>
Trägheitsmoment des Körpers um den Drehpunkt $P$	$\Theta_b$	1,584 · 10 <sup>-3</sup> kg m <sup>2</sup>
Trägheitsmoment des Rades um die Motorachse $M$	$\Theta_w$	0,5375 · 10 <sup>-3</sup> kg m <sup>2</sup>

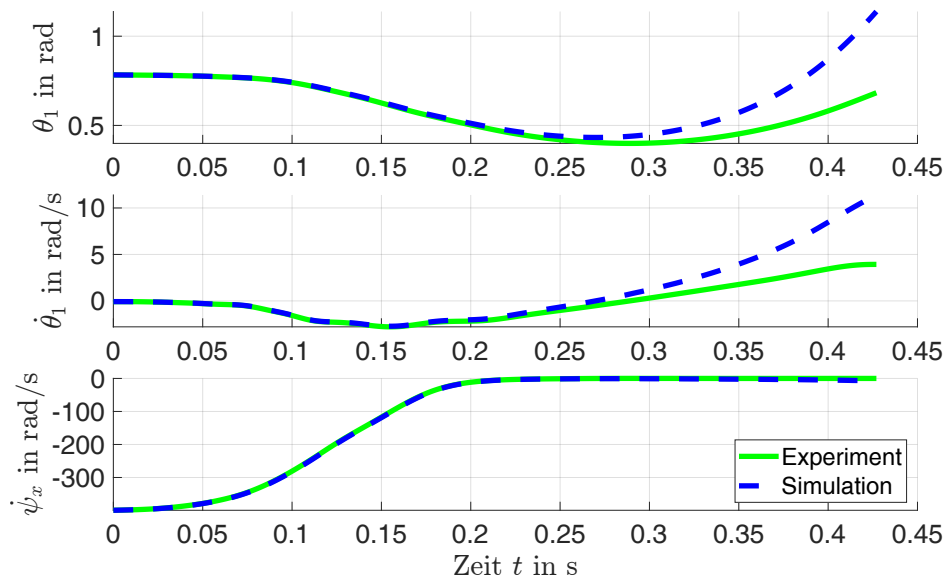
**Tabelle 6.4** Parameter des Prototypens in Gleichung (6.49)

**Hybride Optimalsteuerung** Ziel der hybriden Optimalsteuerung am Würfel ist es, den Aufschwung des Körpers auf die Kante zu bestimmen. Die Stabilisierung durch die Regelung wird hier nicht behandelt. Basierend auf den Methoden aus Unterabschnitt 4.4.1 wird ein hybrides Modell entwickelt, welches für die hybride Optimalsteuerung genutzt werden soll. Mithilfe von Experimenten am Prüfstand wurden anschließend die prädictierten Verläufe mit denen der aufgenommenen Messungen verglichen. Die Ergebnisse sind in den Abbildungen 6.20 sowie 6.21 darge-

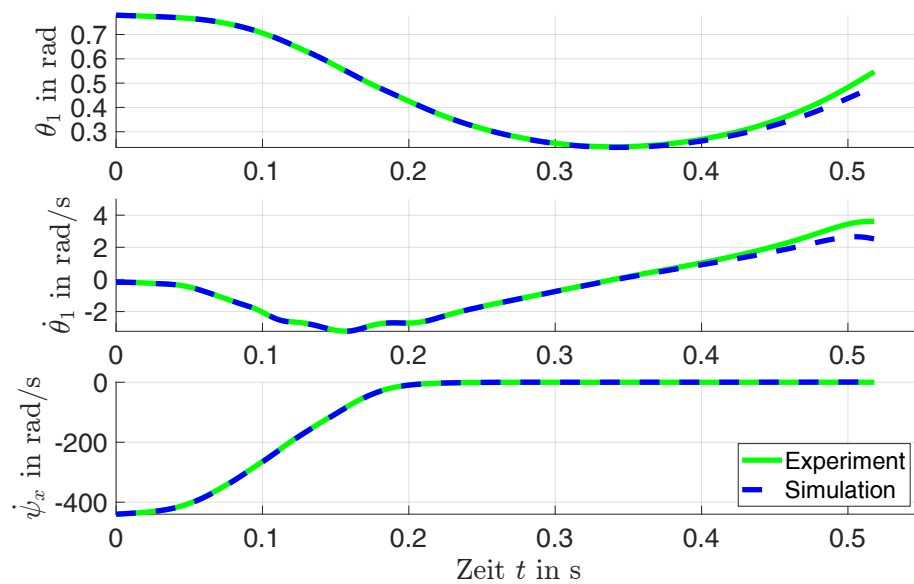
stellt. Beide Abbildungen zeigen einen Aufschwungversuch auf die Kante anhand der Trajektorien der Zustände. Erkennbar ist, dass das Modellverhalten trotz gleicher Anfangsbedingungen sehr unterschiedlich ist. In der Abbildung 6.21 entspricht das Modell der Realität und prädiziert den Aufschwung des Würfels korrekt. Im Gegensatz dazu bestehen in der Abbildung 6.20 große Abweichungen zwischen dem tatsächlich gemessenen und vom Modell berechneten Körperwinkel  $\theta_1$ . Gleiches gilt für die Geschwindigkeit dieses Winkels. Ausschließlich die Winkelgeschwindigkeit  $\dot{\psi}_1$  des Rades stimmt überein. Folgende Punkte müssen im Rahmen der Ursachenforschung und Verbesserung der Qualität weitergehend untersucht werden:

- Der Bremsvorgang zum abrupten Abbremsen des Rades für den Aufschwung wurde ideal modelliert. Aufgrund der Wirkung hoher Kräfte in kurzer Zeit ist das Bremsen nicht reproduzierbar, sondern zeitvariant zu verstehen. Der Bremsvorgang wird u. a. beeinflusst durch eine Abnutzung des Bremsbelags, unterschiedliche Oberflächenbeschaffenheiten entlang der Radoberfläche und einer geringen Unwucht des Rades.
- Da der Würfel aus der oberen Ruhelage sowohl nach links als auch nach rechts fallen kann, wirkt dies bei sehr kleiner Geschwindigkeit wie eine Singularität, weil der Würfel vermeintlich willkürlich zu einer Seite fällt. Die Fallrichtung ist dabei nicht einfach vorherzusagen bzw. zu lernen.
- Möglicherweise ist die angenommene Kovarianzfunktion, welche die Gauß-Prozess-Regression nutzt, nicht geeignet für diese Art von Daten, vgl. Unterabschnitt 3.5.1 und Unterabschnitt 4.4.1. Daher könnten andere Kernel-funktionen zu einer Verbesserung der hybriden Optimalsteuerung führen.

Die Anwendung der hybriden Optimalsteuerung auf den Würfel zeigt die prinzipielle Funktionsfähigkeit der Methode, aber auch wie stark die Wahl geeigneter Messdaten und sinnvoller Hyperparameter den Lernerfolg beeinflusst.



**Abbildung 6.20** Akzeptable Modellgenauigkeit beim Aufschwung durch hybride Optimalsteuerung



**Abbildung 6.21** Hohe Modellgenauigkeit beim Aufschwung durch hybride Optimalsteuerung

# Stichwortverzeichnis

- Anwendungen, 141
  - Doppelpendel, 97, 98
  - Doppelpendel auf einem Wagen, 95, 101
  - Duffing-Oszillator, 120, 134
  - Einfachpendel, 11, 105, 106
  - Einfachpendel auf einem Wagen, 130
  - Golfroboter, 44, 50, 55, 83, 123, 141
  - Servoventil, 50
  - Windenergieanlage, 124
  - Würfel, 159
- Beobachtbarkeit, 109
- Bibliothek, 14, 43
  - Observable, 83
- Carleman-Linearisierung, 27
- DMD, 42
- Dünnbesetztheit, 15, 43, 113, 116, 119
- Echtzeitfähigkeit, 23
- EDMD, 83
- Eigenvektor, 133
- Eigenwert, 133
- Erwartungswert, 70, 100, 110
- Exploration-Exploitation-Dilemma, 19, 91
- Extrapolation, 24
- Fehler
  - Schätzfehler, 110
- Fouriertransformation, 130
- Gauß-Prozess, 71, 104
- Generalisierbarkeit, 18
- Güte
  - Modellgüte, 45, 50, 56
  - Schätzungsgüte, 121
- Hyperparameter, 21, 71
- Hyperparameteroptimierung, 21
- Identifikation
  - Parameteridentifikation, 135
- Inbetriebnahme, 94, 107
- Interpretierbarkeit, 15, 23
- Joint Estimation, 112
  - Augmentiertes Modell, 112, 119
- KI, 16
- Klassifikation, 18
- Konvexität, 85, 115
- Koopman-Operator, 26, 83
- Kovarianz, 100, 110
  - Kovarianzfunktion, 71, 104
  - SE-Kernel, 71, 100
- LASSO-Verfahren, 43, 115
- Least-Squares-Methode, 122
- Linearkombination, 43
- Mittelwertfunktion, 71, 104
- ML, 16
- Modell
  - Augmentiertes Modell, 112, 119
  - Black-Box-Modell, 7, 13
  - Gray-Box-Modell, 7, 14
  - hybrides Modell, 7, 11, 76
  - Modell-Realitäts-Lücke, 45, 109
  - Modelladaption, 132, 135
  - Modellaktualisierung, 132
  - Modellgüte, 45, 50, 56
  - Modellungenauigkeit, 109, 122
  - regelungstechnisch verwertbares Modell, 22
  - Störmodell, 127
  - White-Box-Modell, 7
- Modellierung, 6

- Modellierungsaufwand, 22
- MPC, 82
- Neuronale Netze, 11, 45, 50, 52, 56, 82
  - MOPGRNN, 55, 56
  - PGNN, 47, 51, 53, 57
  - PGRNN, 52, 56, 57
- Normalverteilung, 75, 92, 101, *siehe* Gauß-Verteilung
- Optimierung, 82, 105
  - Bayessche Optimierung, 50, 103, 106, 107
  - Hyperparameteroptimierung, 49, 63
  - Mehrzieloptimierung, 49, 55, 71
    - gewichtete Summe, 49
  - Gütevektroptimierung, 55
  - Optimalsteuerung, 91, 92, 95, 97–99, 168
  - Regularisierung, 22
- Over-/Underfitting, 20, 47
- Passivität, 58
- PCA, 22, 132
- Regelungsentwurf, 22
- Regression, 18
  - Gauß-Prozess-Regression, 70, 72, 91, 100, 103
- Regularisierung, 21
- Reinforcement Learning, 19
- Schätzgüte, 110
- SINDy, 11, 14, 42, 50, 52, 82
- System
  - eingangsaffin, 59, 99, 103, 112, 128
  - PCHD-System, 59, 168
  - Systemveränderung, 23, 123
- Unscented Transformation, 73, 117
- Varianz, 70, 133
- Verteilung
  - Gauß-Verteilung, 118
  - Laplace-Verteilung, 118
  - Posterior-Verteilungen, 101
  - Prior-Verteilung, 100
  - RHS-Verteilung, 118
- Vorverarbeitung, 20, 133
- Vorwissen, 23, 44, 49, 59, 75, 103, 106, 113, 127
- Zustandsbeobachter, 109
  - Kalman-Filter, 110
  - lernfähiger Beobachter, 110
  - Luenberger-Beobachter, 110, 152
  - Sliding-Mode-Beobachter, 111, 129
  - Unscented-Kalman-Filter, 110
  - SRUKF, 114, 118, 120
- Zustandslinearisierung, 99
- Zwei-Freiheitsgrade Regelungsstruktur, 93

# Literatur

- [1] R. Isermann und M. Münchhof, *Identification of Dynamic Systems*. Berlin, Heidelberg: Springer, 2011.
- [2] R. Kruse, C. Borgelt, F. Klawonn, C. Moewes, M. Steinbrecher und P. Held, *Computational Intelligence: A Methodological Introduction*. London: Springer, 2013.
- [3] Bundesverband Informationswirtschaft, Telekommunikation und neue Medien e. V., Deutsches Forschungszentrum für Künstliche Intelligenz GmbH, Hrsg., *Künstliche Intelligenz: Wirtschaftliche Bedeutung, gesellschaftliche Herausforderungen, menschliche Verantwortung*, 2017.
- [4] C. F. Gethmann u. a., *Künstliche Intelligenz in der Forschung: Neue Möglichkeiten und Herausforderungen für die Wissenschaft* (Ethics of science and technology assessment). Berlin und Heidelberg: Springer, 2022, Bd. 48.
- [5] A. Lindholm, N. Wahlström, F. Lindsten und T. B. Schön, *Machine learning: A first course for engineers and scientists*. Cambridge und New York: Cambridge University Press, 2022.
- [6] L. Fahrmeir, T. Kneib, S. Lang und B. D. Marx, *Regression: Models, methods and applications*. Berlin und Heidelberg: Springer, 2021.
- [7] C. M. Bishop und H. Bishop, *Deep learning: Foundations and concepts*. Cham, Schweiz: Springer, 2024.
- [8] S. K. Singh, S. Kumar und P. S. Mehra, „Chat GPT & Google Bard AI: A Review,“ in *International Conference on IoT 2023*.
- [9] R. S. Sutton und A. Barto, *Reinforcement Learning: An introduction* (A Bradford book), Zweite Auflage. Cambridge, Massaschusetts: The MIT Press, 2018.
- [10] C. M. Bishop, *Pattern Recognition and Machine Learning* (Information Science and Statistics). New York: Springer, 2006.
- [11] K. P. Murphy, *Machine Learning: A Probabilistic Perspective* (Adaptive computation and machine learning series). Cambridge: MIT Press, 2012.
- [12] S. Matzka, *Künstliche Intelligenz in den Ingenieurwissenschaften: Maschinelles Lernen verstehen und bewerten*. Wiesbaden und Heidelberg: Springer Vieweg, 2021.
- [13] J. Bergstra, R. Bardenet, Y. Bengio und B. Kégl, „Algorithms for Hyper-Parameter Optimization,“ in *Advances in Neural Information Processing Systems*, J. Shawe-Taylor, R. Zemel, P. Bartlett, F. Pereira und K.Q. Weinberger, Hrsg., Bd. 24, Curran Associates, 2011.

- [14] T. Hastie, R. Tibshirani und M. Wainwright, *Statistical Learning with Sparsity: The Lasso and Generalizations*. New York: Chapman and Hall, 2015.
- [15] I. T. Jolliffe, *Principal Component Analysis* (Springer series in statistics), Zweite Auflage. New York, Berlin und Heidelberg: Springer, 2002.
- [16] A. Junker, „Datengetriebene Modellbildung für nichtlineare mechatronische Systeme in regelungstechnisch verwertbarer Form,“ Dissertation, Universität Paderborn, Paderborn, 2024.
- [17] R.-S. Götte, „Online-Schätzung von Modellungenauigkeiten zur automatischen Modelladaption unter Beibehaltung einer physikalisch-technischen Interpretierbarkeit,“ Dissertation, Universität Paderborn, Paderborn, 2024.
- [18] M. Hesse, „Interaktive Inbetriebnahme von Steuerungen und Regelungen für partiell bekannte dynamische Systeme mittels Gauß-Prozess-Regression,“ Dissertation, Universität Paderborn, Paderborn, 2024.
- [19] T. Schmüdderich, M. Lochbichler und A. Trächtler, „Methodik zur anforderungsgerechten Wahl der Modellierungstiefe von Verhaltensmodellen für die virtuelle Inbetriebnahme,“ in *Fachtagung Mechatronik*, T. Bertram, Hrsg., Aachen: VDI, 2013.
- [20] T. Friebe, M. Schneider, J. Gausemeier und A. Trächtler, „Virtuelle Inbetriebnahme mit wählbarer Modellierungstiefe,“ *Zeitschrift für wirtschaftlichen Fabrikbetrieb*, Jg. 110, Nr. 4, S. 227–232, 2015.
- [21] J. Adamy, *Nichtlineare Systeme und Regelungen*. Berlin, Heidelberg: Springer, 2018.
- [22] A. Karpatne u. a., „Theory-Guided Data Science: A New Paradigm for Scientific Discovery from Data,“ *IEEE Transactions on Knowledge and Data Engineering*, Jg. 29, Nr. 10, S. 2318–2331, 2017.
- [23] G. E. Karniadakis, I. G. Kevrekidis, L. Lu, P. Perdikaris, S. Wang und L. Yang, „Physics-informed machine learning,“ *Nature Reviews Physics*, Jg. 3, Nr. 6, S. 422–440, 2021.
- [24] O. Föllinger, U. Konigorski, B. Lohmann, G. Roppenecker und A. Trächtler, *Regelungstechnik: Einführung in die Methoden und ihre Anwendung*, 12. überarbeitete Auflage. Berlin und Offenbach: VDE Verlag GmbH, 2016.
- [25] A. Junker, J. Timmermann und A. Trächtler, „Data-Driven Models for Control Engineering Applications Using the Koopman Operator,“ in *3rd International Conference on Artificial Intelligence, Robotics and Control (AIRC)*, IEEE, 2022.
- [26] B. O. Koopman, „Hamiltonian Systems and Transformation in Hilbert Space,“ *Proceedings of the National Academy of Sciences of the United States of America*, Jg. 17, Nr. 5, S. 315–318, 1931.
- [27] S. L. Brunton, B. W. Brunton, J. L. Proctor und J. N. Kutz, „Koopman Invariant Subspaces and Finite Linear Representations of Nonlinear Dynamical Systems for Control,“ *PloS one*, Jg. 11, Nr. 2, 2016.
- [28] S. L. Brunton und J. N. Kutz, *Data-Driven Science and Engineering*. Cambridge University Press, 2019, Bd. 237.



- [29] T. Carleman, „Application de la théorie des équations intégrales linéaires aux systèmes d'équations différentielles non linéaires,“ *Acta Math*, Jg. 59, S. 63–87, 1932.
- [30] A. Amini, Q. Sun und N. Motee, „Carleman State Feedback Control Design of a Class of Nonlinear Control Systems,“ *IFAC-PapersOnLine*, Jg. 52, Nr. 20, S. 229–234, 2019.
- [31] I. Mezić, „Spectral Properties of Dynamical Systems, Model Reduction and Decompositions,“ *Nonlinear Dynamics*, Jg. 41, Nr. 1-3, S. 309–325, 2005.
- [32] P. J. Schmid und J. Sesterhenn, *Dynamic Mode Decomposition of numerical and experimental data: Vortrag bei American Physical Society*, 2008.
- [33] P. J. Schmid, „Dynamic mode decomposition of numerical and experimental data,“ *Journal of Fluid Mechanics*, Jg. 656, S. 5–28, 2010.
- [34] G. Berkooz, P. Holmes und J. L. Lumley, „The Proper Orthogonal Decomposition in the Analysis of Turbulent Flows,“ *Annual Review of Fluid Mechanics*, Jg. 25, Nr. 1, S. 539–575, 1993.
- [35] A. Antoulas und D. Sorensen, „Approximation of large-scale dynamical systems: an overview,“ *International Journal of Applied Mathematics and Computer Science*, Jg. 11, Nr. 5, S. 1093–1121, 2001.
- [36] Y. Saad, „Variations on Arnoldi's Method for Computing Eigenvalues of Large Unsymmetric Matrices,“ *Linear Algebra and its Applications*, Jg. 34, S. 269–295, 1980.
- [37] C. W. Rowley, I. Mezić, S. Bagherie, P. Schlatter und D. S. Henningson, „Spectral analysis of nonlinear flows,“ *Journal of Fluid Mechanics*, Jg. 641, S. 115–127, 2009.
- [38] O. Pust und P. J. Schmid, „Stabilitätsanalyse experimenteller Daten mit der Dynamic Mode Decomposition,“ *Fachtagung Lasermethoden in der Strömungsmesstechnik, Erlangen*, 2009.
- [39] J. H. Tu, C. W. Rowley, D. M. Luchtenburg, S. L. Brunton und J. N. Kutz, „On Dynamic Mode Decomposition: Theory and Applications,“ *Journal of Computational Dynamics*, Jg. 1, Nr. 2, S. 391–421, 2014.
- [40] J. L. Proctor, S. L. Brunton und J. N. Kutz, „Dynamic Mode Decomposition with Control,“ *SIAM Journal on Applied Dynamical Systems*, Jg. 15, Nr. 1, S. 142–161, 2016.
- [41] P. J. Baddoo, B. Herrmann, B. J. McKeon, J. Nathan Kutz und S. L. Brunton, „Physics-informed dynamic mode decomposition,“ *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, Jg. 479, Nr. 2271, 2023.
- [42] N. Gillis, M. Karow und P. Sharma, „Approximating the nearest stable discrete-time system,“ *Linear Algebra and its Applications*, Jg. 573, S. 37–53, 2019.
- [43] N. Gillis und P. Sharma, „On computing the distance to stability for matrices using linear dissipative Hamiltonian systems,“ *Automatica*, Jg. 85, Nr. 2, S. 113–121, 2017.

- [44] M. O. Williams, I. G. Kevrekidis und C. W. Rowley, „A Data-Driven Approximation of the Koopman Operator: Extending Dynamic Mode Decomposition,“ *Journal of Nonlinear Science*, Jg. 25, Nr. 6, S. 1307–1346, 2015.
- [45] G. Mamakoukas, M. L. Castano, X. Tan und T. D. Murphey, „Local Koopman Operators for Data-Driven Control of Robotic Systems,“ in *Robotics: Science and Systems XV*, Robotics: Science and Systems Foundation, 2019.
- [46] M. O. Williams, M. S. Hemati, S. T. M. Dawson, I. G. Kevrekidis und C. W. Rowley, „Extending Data-Driven Koopman Analysis to Actuated Systems,“ *IFAC-PapersOnLine*, Jg. 49, Nr. 18, S. 704–709, 2016.
- [47] M. Korda und I. Mezić, „Linear predictors for nonlinear dynamical systems: Koopman operator meets model predictive control,“ *Automatica*, Jg. 93, S. 149–160, 2018.
- [48] G. Mamakoukas, I. Abraham und T. D. Murphey, „Learning Stable Models for Prediction and Control,“ *IEEE Transactions on Robotics*, Jg. 39, Nr. 3, S. 2255–2275, 2023.
- [49] J. L. Proctor, S. L. Brunton und J. N. Kutz, „Generalizing Koopman Theory to Allow for Inputs and Control,“ *SIAM Journal on Applied Dynamical Systems*, Jg. 17, Nr. 1, S. 909–930, 2018.
- [50] L. Shi und K. Karydis, „ACD-EDMD: Analytical Construction for Dictionaries of Lifting Functions in Koopman Operator-Based Nonlinear Robotic Systems,“ *IEEE Robotics and Automation Letters*, Jg. 7, Nr. 2, S. 906–913, 2022.
- [51] M. O. Williams, C. W. Rowley und I. G. Kevrekidis, „A Kernel-based Method for Data-driven Koopman Spectral Analysis,“ *Journal of Computational Dynamics*, Jg. 2, Nr. 2, S. 247–265, 2015.
- [52] Q. Li, F. Dietrich, E. M. Bollt und I. G. Kevrekidis, „Extended dynamic mode decomposition with dictionary learning: a data-driven adaptive spectral decomposition of the Koopman operator,“ *Chaos: An Interdisciplinary Journal of Nonlinear Science*, Jg. 27, Nr. 10, 2017.
- [53] I. Mezić, *Operator is the Model*, arXiv Preprint, arXiv:2310.18516, 2023.
- [54] M. Netto, Y. Susuki, V. Krishnan und Y. Zhang, „On Analytical Construction of Observable Functions in Extended Dynamic Mode Decomposition for Nonlinear Estimation and Prediction,“ *IEEE Control Systems Letters*, Jg. 5, Nr. 6, S. 1868–1873, 2021.
- [55] G. Mamakoukas, M. L. Castano, X. Tan und T. D. Murphey, „Derivative-Based Koopman Operators for Real-Time Control of Robotic Systems,“ *IEEE Transactions on Robotics*, Jg. 37, Nr. 6, S. 2173–2192, 2021.
- [56] M. L. Castano, A. Hess, G. Mamakoukas, T. Gao, T. D. Murphey und X. Tan, „Control-oriented Modeling of Soft Robotic Swimmer with Koopman Operators,“ in *2020 IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM)*, IEEE, 2020, S. 1679–1685.

- [57] S. Hanke, S. Peitz, O. Wallscheid, S. Klus, J. Böcker und M. Dellnitz, *Koopman Operator Based Finite-Set Model Predictive Control for Electrical Drives*, arXiv Preprint, arXiv:1804.00854, 2018.
- [58] S. Peitz, *Controlling nonlinear PDEs using low-dimensional bilinear approximations obtained from data*, arXiv Preprint, arXiv:180106419, 2018.
- [59] S. Peitz und S. Klus, „Koopman operator-based model reduction for switched-system control of PDEs,“ *Automatica*, Jg. 106, S. 184–191, 2019.
- [60] S. Peitz und S. Klus, „Feedback Control of Nonlinear PDEs Using Data-Efficient Reduced Order Models Based on the Koopman Operator,“ in *The Koopman Operator in Systems and Control*, Ser. Lecture Notes in Control and Information Sciences, A. Mauroy, I. Mezić und Y. Susuki, Hrsg., Bd. 484, Cham: Springer International Publishing, 2020, S. 257–282.
- [61] S. Peitz, S. E. Otto und C. W. Rowley, „Data-Driven Model Predictive Control using Interpolated Koopman Generators,“ *SIAM Journal on Applied Dynamical Systems*, Jg. 19, Nr. 3, S. 2162–2193, 2020.
- [62] S. Peitz und K. Bieker, „On the universal transformation of data-driven models to control systems,“ *Automatica*, Jg. 149, 2023.
- [63] M. Bonnert, „Identifikation, Analyse und Regelung nichtlinearer dynamischer Systeme mittels des Koopman-Operators, künstlicher neuronaler Netze und linearer Zustandsregler,“ Dissertation, Technische Universität Darmstadt, Darmstadt, 2022.
- [64] G. Mamakoukas, O. Xherija und T. D. Murphey, „Memory-Efficient Learning of Stable Linear Dynamical Systems for Prediction and Control,“ in *Advances in Neural Information Processing Systems*, H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan und H. Lin, Hrsg., Bd. 33, Curran Associates, 2020, S. 13 527–13 538.
- [65] J. L. Hennessy und D. A. Patterson, *Computer Architecture: A Quantitative Approach*, Sechste Auflage. Cambridge, Massachusetts: William Andrew Publishing, 2019.
- [66] Agner Fog, *The microarchitecture of Intel, AMD, and VIA CPUs: An optimization guide for assembly programmers and compiler makers*, <https://www.agner.org/optimize/microarchitecture.pdf> (Stand Dezember 2024), 2023.
- [67] K. Pape, „Analyse des Rechenaufwands der numerischen Approximationsverfahren des Koopman-Operators,“ Studienarbeit, Universität Paderborn, Paderborn, 2022.
- [68] H. Wörn und U. Brinkschulte, *Echtzeitsysteme: Grundlagen, Funktionsweisen, Anwendungen; mit 32 Tabellen*. Berlin, Heidelberg: Springer, 2005.
- [69] R. Götte, „Modellprädiktive Regelung des Pendels mittels einer linear approximierten Zustandsbeschreibung basierend auf dem Koopman-Operator,“ Masterarbeit (Technomathematik), Universität Paderborn, Paderborn, 2019.
- [70] A. Junker, K. Pape, J. Timmermann und A. Trächtler, „Adaptive Koopman-Based Models for Holistic Controller and Observer Design,“ *IFAC-PapersOnLine*, Jg. 56, Nr. 3, S. 625–630, 2023.

- [71] H. M. Calderón, E. Schulz, T. Oehlschlägel und H. Werner, „Koopman Operator-based Model Predictive Control with Recursive Online Update,“ in *2021 European Control Conference (ECC)*, Piscataway, NJ: IEEE, 2021, S. 1543–1549.
- [72] S. L. Brunton, J. L. Proctor und J. N. Kutz, „Discovering governing equations from data by sparse identification of nonlinear dynamical systems,“ *Proceedings of the National Academy of Sciences of the United States of America*, Jg. 113, Nr. 15, S. 3932–3937, 2016.
- [73] R. Tibshirani, „Regression Shrinkage and Selection Via the Lasso,“ *Journal of the Royal Statistical Society: Series B*, Jg. 58, Nr. 1, S. 267–288, 1996.
- [74] S. L. Brunton, J. L. Proctor und J. N. Kutz, „Sparse Identification of Nonlinear Dynamics with Control (SINDYc),“ *IFAC-PapersOnLine*, Jg. 49, Nr. 18, S. 710–715, 2016.
- [75] T. Chen und H. Chen, „Approximation Capability to Functions of Several Variables, Nonlinear Functionals, and Operators by Radial Basis Function Neural Networks,“ *IEEE Transactions on Neural Networks*, Jg. 6, Nr. 4, S. 904–910, 1995.
- [76] T. Chen und H. Chen, „Universal Approximation to Nonlinear Operators by Neural Networks with Arbitrary Activation Functions and Its Application to Dynamical Systems,“ *IEEE Transactions on Neural Networks*, Jg. 6, Nr. 4, S. 911–917, 1995.
- [77] K.-i. Funahashi und Y. Nakamura, „Approximation of dynamical systems by continuous time recurrent neural networks,“ *Neural Networks*, Jg. 6, Nr. 6, S. 801–806, 1993.
- [78] K. Hornik, M. Stinchcombe und H. White, „Multilayer Feedforward Networks are Universal Approximators,“ *Neural Networks*, Jg. 2, S. 359–366, 1989.
- [79] A. Daw, A. Karpatne, W. D. Watkins, J. S. Read und V. Kumar, „Physics-Guided Neural Networks (PGNN): An Application in Lake Temperature Modeling,“ in *Knowledge-Guided Machine Learning*, A. Karpatne, R. Kannan und V. Kumar, Hrsg., Boca Raton: Chapman and Hall, 2022, S. 353–372.
- [80] J. Pathak u. a., „Hybrid forecasting of chaotic processes: Using machine learning in conjunction with a knowledge-based model,“ *Chaos: An Interdisciplinary Journal of Nonlinear Science*, Jg. 28, Nr. 4, 2018.
- [81] D. C. Psichogios und L. H. Ungar, „A hybrid neural network-first principles approach to process modeling,“ *AIChE Journal*, Jg. 38, Nr. 10, S. 1499–1511, 1992.
- [82] A. Kroll, *Computational Intelligence: Probleme, Methoden und technische Anwendungen*, Zweite Auflage. Berlin und Boston: De Gruyter Oldenbourg, 2016.
- [83] D. P. Kingma und J. Ba, „Adam: A Method for Stochastic Optimization,“ in *3rd International Conference on Learning Representations (ICLR), San Diego, USA*, Yoshua Bengio und Yann LeCun, Hrsg., 2015.

- [84] R.-S. Götte und J. Timmermann, „Composed Physics- and Data-driven System Identification for Non-autonomous Systems in Control Engineering,“ *3rd International Conference on Artificial Intelligence, Robotics and Control (AIRC)*, Jg. 38, S. 67–76, 2022.
- [85] N. Muralidhar, M. R. Islam, M. Marwah, A. Karpatne und N. Ramakrishnan, „Incorporating Prior Domain Knowledge into Deep Neural Networks,“ in *IEEE International Conference on Big Data*, IEEE, 2018, S. 36–45.
- [86] S. J. Raymond und D. B. Camarillo, *Applying physics-based loss functions to neural networks for improved generalizability in mechanics problems*, arXiv Preprint, arXiv:2105.00075, 2021.
- [87] M. Ehrgott, *Multicriteria Optimization: With 88 figures and 12 tables*, 2. Auflage. Berlin, Heidelberg und New York: Springer, 2005, Bd. 491.
- [88] J. Nocedal und S. J. Wright, *Numerical Optimization* (Springer Series in Operations Research and Financial Engineering), 2. Auflage. New York: Springer, 2006.
- [89] F. M. Rohrhofer, S. Posch, C. Gößnitzer und B. C. Geiger, „Data vs. Physics: The Apparent Pareto Front of Physics-Informed Neural Networks,“ *IEEE Access*, Jg. 11, S. 86 252–86 261, 2023.
- [90] M. Lutter, C. Ritter und J. Peters, „Deep Lagrangian Networks: Using Physics as Model Prior for Deep Learning,“ in *7th International Conference on Learning Representations (ICLR)*, 2019.
- [91] S. Greydanus, M. Dzamba und J. Yosinski, „Hamiltonian neural networks,“ in *Proceedings of the 33rd International Conference on Neural Information Processing Systems*, Red Hook, NY, USA: Curran Associates Inc, 2019.
- [92] M. Raissi, P. Perdikaris und G. E. Karniadakis, „Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations,“ in *Journal of Computational Physics*, Bd. 378, 2019, S. 686–707.
- [93] E. A. Antonelo, E. Camponogara, L. O. Seman, J. P. Jordanou, E. R. de Souza und J. F. Hübner, „Physics-informed neural nets for control of dynamical systems,“ *Neurocomputing*, Jg. 579, 2024.
- [94] J. Nicodemus, J. Kneifl, J. Fehr und B. Unger, „Physics-informed Neural Networks-based Model Predictive Control for Multi-link Manipulators,“ *IFAC-PapersOnLine*, Jg. 55, Nr. 20, S. 331–336, 2022.
- [95] O. Schön, R.-S. Götte und J. Timmermann, „Multi-Objective Physics-Guided Recurrent Neural Networks for Identifying Non-Autonomous Dynamical Systems,“ *IFAC-PapersOnLine*, Jg. 55, Nr. 12, S. 19–24, 2022.
- [96] L. Kersting, „Nichtlineare Ventilmodellierung,“ Interne Dokumentation, Universität Paderborn, 2021.
- [97] M. Lochbichler, „Systematische Wahl einer Modellierungstiefe im Entwurfsprozess mechatronischer Systeme,“ Dissertation, Universität Paderborn, Paderborn, 2020.

- [98] X. Jia u. a., „Physics-Guided Recurrent Graph Model for Predicting Flow and Temperature in River Networks,“ in *Proceedings of the 2021 SIAM International Conference on Data Mining (SDM)*, C. Demeniconi und I. Davidson, Hrsg., Society for Industrial and Applied Mathematics, 2021, S. 612–620.
- [99] S. Hochreiter und J. Schmidhuber, „Long Short-Term Memory,“ *Neural computation*, Jg. 9, Nr. 8, S. 1735–1780, 1997.
- [100] K. Cho u. a., „Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation,“ in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*, A. Moschitti, B. Pang und W. Daelemans, Hrsg., Association for Computational Linguistics, 2014, S. 1724–1734.
- [101] G. Kreisselmeier und R. Steinhauser, „Systematic Control Design by Optimizing a Vector Performance Index,“ *IFAC Proceedings Volumes*, Jg. 12, Nr. 7, S. 113–117, 1979.
- [102] A. Junker, J. Timmermann und A. Trächtler, „Learning Data-Driven PCHD Models for Control Engineering Applications,“ *IFAC-PapersOnLine*, Jg. 55, Nr. 12, S. 389–394, 2022.
- [103] R. E. Kalman und J. E. Bertram, „Control System Analysis and Design Via the “Second Method” of Lyapunov: I - Continuous-Time Systems,“ *Journal of Basic Engineering*, Jg. 82, Nr. 2, S. 371–393, 1960.
- [104] J. C. Willems, „Dissipative dynamical systems part I: General theory,“ *Archive for Rational Mechanics and Analysis*, Jg. 45, Nr. 5, S. 321–351, 1972.
- [105] V.-M. Popov, „The solution of a new stability problem for controlled systems,“ *Avtomatika i Telemekhanika*, S. 7–26, 1963.
- [106] B. D. Anderson, „A simplified viewpoint of hyperstability,“ *IEEE Transactions on Automatic Control*, Jg. 13, Nr. 3, S. 292–294, 1968.
- [107] V.-M. Popov, *Hyperstability of control systems*. Bukarest: Editura Academiei, 1973, Bd. 204.
- [108] C. I. Byrnes, A. Isidori und J. C. Willems, „Passivity, feedback equivalence, and the global stabilization of minimum phase nonlinear systems,“ *IEEE Transactions on Automatic Control*, Jg. 36, Nr. 11, S. 1228–1240, 1991.
- [109] H. K. Khalil, *Nonlinear control*. Boston: Pearson, 2015.
- [110] B. Maschke und A. van der Schaft, „Port-Controlled Hamiltonian Systems: Modelling Origins and Systemtheoretic Properties,“ *IFAC Proceedings Volumes*, Jg. 25, Nr. 13, S. 359–365, 1992.
- [111] B. Maschke, A. van der Schaft und P. C. Breedveld, „An intrinsic hamiltonian formulation of network dynamics: non-standard poisson structures and gyrators,“ *Journal of the Franklin Institute*, Jg. 329, Nr. 5, S. 923–966, 1992.
- [112] A. van der Schaft, „Port-Hamiltonian Modeling for Control,“ *Annual Review of Control, Robotics, and Autonomous Systems*, Jg. 3, Nr. 1, S. 393–416, 2020.
- [113] A. van der Schaft und D. Jeltsema, *Port-Hamiltonian systems theory: An introductory overview*. Boston und Delft: Now, 2014, Bd. 1, 2/3 (2014).

- [114] B. Maschke, R. Ortega und A. van der Schaft, „Energy-based Lyapunov functions for forced Hamiltonian systems with dissipation,“ *IEEE Transactions on Automatic Control*, Jg. 45, Nr. 8, S. 1498–1502, 2000.
- [115] P. Cvitanović, R. Artuso, R. Mainieri, G. Tanner und G. Vattay, *Chaos: Classical and Quantum*. Copenhagen: Niels Bohr Institute, 2022.
- [116] J. Timmermann, S. Khatab, S. Ober-Blöbaum und A. Trächtler, „Discrete Mechanics and Optimal Control and its Application to a Double Pendulum on a Cart,“ *IFAC Proceedings Volumes*, Jg. 44, Nr. 1, S. 10 199–10 206, 2011.
- [117] I. Abraham, G. de La Torre und T. D. Murphey, „Model-Based Control Using Koopman Operators,“ in *Robotics: Science and Systems XIII*, Robotics: Science and Systems Foundation, 2017.
- [118] Q. Quan, *Introduction to Multicopter Design and Control*. Singapore: Springer Singapore, 2017.
- [119] V. Gatteschi u. a., „New Frontiers of Delivery Services Using Drones: A Prototype System Exploiting a Quadcopter for Autonomous Drug Shipments,“ in *2015 IEEE 39th Annual Computer Software and Applications Conference*, IEEE, 2015, S. 920–927.
- [120] B. Rabta, C. Wankmüller und G. Reiner, „A drone fleet model for last-mile distribution in disaster relief operations,“ *International Journal of Disaster Risk Reduction*, Jg. 28, S. 107–112, 2018.
- [121] P. Martin, S. Devasia und B. Paden, „A different look at output tracking: control of a vtol aircraft,“ *Automatica*, Jg. 32, Nr. 1, S. 101–107, 1996.
- [122] M. Fliess, J. Levine, P. Martin und P. Rouchon, „A Lie-Backlund approach to equivalence and flatness of nonlinear systems,“ *IEEE Transactions on Automatic Control*, Jg. 44, Nr. 5, S. 922–937, 1999.
- [123] A. Junker, J. Timmermann und A. Trächtler, „Adaptive Data-Driven Models in Port-Hamiltonian Form for Control Design,“ *PAMM*, Jg. 25, Nr. 1, 2025.
- [124] C. E. Rasmussen und C. K. I. Williams, *Gaussian processes for machine learning* (Adaptive computation and machine learning), 3. Auflage. Cambridge, Mass.: MIT Press, 2008.
- [125] S. Kamthe und M. P. Deisenroth, *Data-Efficient Reinforcement Learning with Probabilistic Model Predictive Control*, arXiv Preprint, arXiv:1706.06491, 2017.
- [126] M. P. Deisenroth, D. Fox und C. E. Rasmussen, „Gaussian Processes for Data-Efficient Learning in Robotics and Control,“ *IEEE transactions on pattern analysis and machine intelligence*, Jg. 37, Nr. 2, S. 408–423, 2015.
- [127] D. Simon, *Optimal state estimation: Kalman,  $H_\infty$ , and nonlinear approaches*. Hoboken, NJ: Wiley-Interscience, 2006.
- [128] A. S. Bazanella, *Data-Driven Controller Design* (Communications and Control Engineering Ser). Dordrecht: Springer, 2011.
- [129] E. Kaiser, J. N. Kutz und S. L. Brunton, „Data-driven discovery of Koopman eigenfunctions for control,“ *Machine Learning: Science and Technology*, Jg. 2, Nr. 3, 2021.

- [130] X. Ma, B. Huang und U. Vaidya, „Optimal Quadratic Regulation of Nonlinear System Using Koopman Operator,“ in *2019 American Control Conference (ACC)*, IEEE, 2019, S. 4911–4916.
- [131] A. Maksakov und S. Palis, „Koopman-based Optimal Control of Boost DC-DC Converter,“ in *2020 IEEE Problems of Automated Electrodrive. Theory and Practice (PAEP)*, IEEE, 2020.
- [132] T. Iwata und Y. Kawahara, „Controlling Nonlinear Dynamical Systems with Linear Quadratic Regulator-based Policy Networks in Koopman space,“ in *2021 60th IEEE Conference on Decision and Control (CDC)*, IEEE, 2021, S. 5086–5091.
- [133] V. Zinage und E. Bakolas, „Koopman Operator Based Modeling and Control of Rigid Body Motion Represented by Dual Quaternions,“ in *2022 American Control Conference (ACC)*, Piscataway, NJ: IEEE, 2022, S. 3997–4002.
- [134] J. S. Kim, Y. S. Quan und C. C. Chung, „Data-Driven Modeling and Control for Lane Keeping System of Automated Driving Vehicles: Koopman Operator Approach,“ in *2022 22nd International Conference on Control, Automation and Systems (ICCAS)*, IEEE, 2022, S. 1049–1055.
- [135] M. Rahmani und S. Redkar, „Optimal control of a MEMS gyroscope based on the Koopman theory,“ *International Journal of Dynamics and Control*, 2023.
- [136] P. Bevanda, M. Beier, S. Heshmati-Alamdari, S. Sosnowski und S. Hirche, „Towards Data-driven LQR with Koopmanizing Flows,“ *IFAC-PapersOnLine*, Jg. 55, Nr. 15, S. 13–18, 2022.
- [137] I. Abraham und T. D. Murphey, „Active Learning of Dynamics for Data-Driven Control Using Koopman Operators,“ *IEEE Transactions on Robotics*, 2019.
- [138] A. Krolicki, D. Rufino und D. Tellez-Castro, „Finite Time Nonlinear Optimal Control using Koopman Eigenfunctions,“ in *2023 American Control Conference (ACC)*, Piscataway, NJ: IEEE, 2023, S. 2117–2122.
- [139] A. J. Gibson, M. L. Calvisi und X. C. Yee, „Koopman Linear Quadratic Regulator Using Complex Eigenfunctions for Nonlinear Dynamical Systems,“ *SIAM Journal on Applied Dynamical Systems*, Jg. 21, Nr. 4, S. 2463–2486, 2022.
- [140] L. Heinemann, „Riccati-Regelungen für nichtlineare Systeme mittels des Koopman-Operators,“ Studienarbeit (Maschinenbau), Universität Paderborn, Paderborn, 2022.
- [141] S. Gros, M. Zanon, R. Quirynen, A. Bemporad und M. Diehl, „From linear to nonlinear MPC: bridging the gap via the real-time iteration,“ *International Journal of Control*, Jg. 93, Nr. 1, S. 62–80, 2020.
- [142] K. M. H. Baunsgaard, O. Ravn, C. S. Kallesoe und N. K. Poulsen, „MPC control of water supply networks,“ in *2016 European Control Conference (ECC)*, IEEE, 2016, S. 1770–1775.



- [143] K. Sunny, A. Sheikh und S. Wagh, „Dynamic Mode Decomposition for Prediction and Enhancement of Rotor Angle Stability,“ in *2020 7th International Conference on Control, Decision and Information Technologies (CoDIT)*, IEEE, 2020, S. 160–165.
- [144] E. Kaiser, J. N. Kutz und S. L. Brunton, „Sparse identification of nonlinear dynamics for model predictive control in the low-data limit,“ *Proceedings of the Royal Society A: Mathematical, physical, and engineering sciences*, Jg. 474, Nr. 2219, 2018.
- [145] A. Afram, F. Janabi-Sharifi, A. S. Fung und K. Raahemifar, „Artificial neural network (ANN) based model predictive control (MPC) and optimization of HVAC systems: A state of the art review and case study of a residential HVAC system,“ *Energy and Buildings*, Jg. 141, S. 96–113, 2017.
- [146] J. Berberich, J. Kohler, M. A. Muller und F. Allgöwer, „Data-Driven Model Predictive Control With Stability and Robustness Guarantees,“ *IEEE Transactions on Automatic Control*, Jg. 66, Nr. 4, S. 1702–1717, 2021.
- [147] D. Bruder, B. Gillespie, C. David Remy und R. Vasudevan, „Modeling and Control of Soft Robots Using the Koopman Operator and Model Predictive Control,“ in *Robotics: Science and Systems XV*, Robotics: Science and Systems Foundation, 2019.
- [148] D. Bruder, X. Fu, R. B. Gillespie, C. D. Remy und R. Vasudevan, „Data-Driven Control of Soft Robots Using Koopman Operator Theory,“ *IEEE Transactions on Robotics*, S. 948–961, 2020.
- [149] J. Wang u. a., „An Improved Koopman-MPC Framework for Data-Driven Modeling and Control of Soft Actuators,“ *IEEE Robotics and Automation Letters*, Jg. 8, Nr. 2, S. 616–623, 2023.
- [150] L. Do, M. Korda und Z. Hurák, „Controlled synchronization of coupled pendulums by Koopman Model Predictive Control,“ *Control Engineering Practice*, Jg. 139, 2023.
- [151] S. Xie und J. Ren, „Discrete System Linearization using Koopman Operators for Predictive Control and Its Application in Nano-positioning,“ in *2020 IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM)*, IEEE, 2020, S. 1374–1379.
- [152] S. Xie und J. Ren, „Linearization of Recurrent-Neural-Network- Based Models for Predictive Control of Nano-Positioning Systems Using Data-Driven Koopman Operators,“ *IEEE Access*, Jg. 8, S. 147 077–147 088, 2020.
- [153] X. Huang und H.-T. Zhang, „Koopman Operator Learning Control of the Piezoelectric Actuator,“ in *2023 42nd Chinese Control Conference (CCC)*, IEEE, 2023, S. 6731–6735.
- [154] C. Folkestad, D. Pastor und J. W. Burdick, „Episodic Koopman Learning of Nonlinear Robot Dynamics with Application to Fast Multirotor Landing,“ in *2020 IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, 2020, S. 9216–9222.

- [155] G. Mamakoukas, S. Di Cairano und A. P. Vinod, „Robust Model Predictive Control with Data-Driven Koopman Operators,“ in *2022 American Control Conference (ACC)*, Piscataway, NJ: IEEE, 2022, S. 3885–3892.
- [156] A. Tabassum und H. Bai, „Data-Driven Koopman-based Modeling and Control of Quadcopter in a Turbulent Wind Field,“ in *2023 IEEE Conference on Control Technology and Applications (CCTA)*, IEEE, 2023, S. 490–495.
- [157] M. Švec, Š. Ileš und J. Matuško, „Predictive Direct Yaw Moment Control Based on the Koopman Operator,“ *IEEE Transactions on Control Systems Technology*, S. 2912–2919, 2023.
- [158] S. Li, Z. Xu, J. Liu und C. Xu, „Learning-based Extended Dynamic Mode Decomposition for Addressing Path-following Problem of Underactuated Ships with Unknown Dynamics,“ *International Journal of Control, Automation and Systems*, S. 4076–4089, 2022.
- [159] B. Chen u. a., „Data-Driven Koopman Model Predictive Control for Optimal Operation of High-Speed Trains,“ *IEEE Access*, Jg. 9, S. 82 233–82 248, 2021.
- [160] M. B. Saltik, B. Jayawardhana und A. Cherukuri, „Iterative Learning and Model Predictive Control for Repetitive Nonlinear Systems via Koopman Operator Approximation,“ in *2022 IEEE 61st Conference on Decision and Control (CDC)*, IEEE, 2022, S. 3059–3065.
- [161] A. Narasingam und J. S.-I. Kwon, „Koopman operator-based model identification and control of hydraulic fracture propagation,“ in *2020 American Control Conference (ACC)*, IEEE, 2020, S. 4533–4538.
- [162] J. Yuan u. a., „Koopman modeling and optimal control for microbial fed-batch fermentation with switching operators,“ *Nonlinear Analysis: Hybrid Systems*, Jg. 52, 2024.
- [163] H. Arbabi, M. Korda und I. Mezić, „A Data-Driven Koopman Model Predictive Control Framework for Nonlinear Partial Differential Equations,“ in *2018 IEEE Conference on Decision and Control (CDC)*, Piscataway, NJ: IEEE, 2018, S. 6409–6414.
- [164] R. Ortega und M. W. Spong, „Adaptive motion control of rigid robots: A tutorial,“ *Automatica*, Jg. 25, Nr. 6, S. 877–888, 1989.
- [165] R. Ortega, A. van der Schaft, B. Maschke und G. Escobar, „Energy-shaping of port-controlled Hamiltonian systems by interconnection,“ in *Proceedings of the 38th IEEE Conference on Decision and Control*, IEEE, 1999, S. 1646–1651.
- [166] R. Ortega, A. van der Schaft, B. Maschke und G. Escobar, „Interconnection and damping assignment passivity-based control of port-controlled Hamiltonian systems,“ *Automatica*, Jg. 38, Nr. 4, S. 585–596, 2002.
- [167] R. Ortega und E. García-Canseco, „Interconnection and Damping Assignment Passivity-Based Control: A Survey,“ *European Journal of Control*, Jg. 10, Nr. 5, S. 432–450, 2004.

- [168] P. Kotyczka und B. Lohmann, „Parametrization of IDA-PBC by assignment of local linear dynamics,“ in *2009 European Control Conference*, Piscataway, New Jersey: IEEE, 2009, S. 4721–4726.
- [169] P. Kotyczka und B. Lohmann, „Parametrierung von IDA-PBC über Zuweisung lokal linearer Dynamik,“ *at - Automatisierungstechnik*, Jg. 58, Nr. 1, 2010.
- [170] S. Sanchez-Escalonilla, R. Reyes-Baez und B. Jayawardhana, „Total Energy Shaping with Neural Interconnection and Damping Assignment - Passivity Based Control,“ in *Proceedings of The 4th Annual Learning for Dynamics and Control Conference*, R. Firoozi u. a., Hrsg., Ser. Proceedings of Machine Learning Research, Bd. 168, PMLR, 2022, S. 520–531.
- [171] VDI/VDE-Gesellschaft Mess- und Automatisierungstechnik, *VDI/VDE 2206: Entwicklung mechatronischer und cyber-physischer Systeme*, 2021.
- [172] J. T. Betts, „Survey of Numerical Methods for Trajectory Optimization,“ *Journal of Guidance, Control, and Dynamics*, Jg. 21, Nr. 2, S. 193–207, 1998.
- [173] C. Geiger und C. Kanzow, *Theorie und Numerik restringierter Optimierungsaufgaben: Mit 140 Übungsaufgaben*. Berlin und Heidelberg: Springer, 2002.
- [174] O. Föllinger, *Optimale Regelung und Steuerung: Mit 7 Tabellen und 16 Übungsaufgaben mit genauer Darstellung des Lösungsweges*, 3. Auflage. München und Wien: Oldenbourg, 1994.
- [175] M. Hesse, J. Timmermann und A. Trächtler, „Hybrid Optimal Control for Dynamical Systems using Gaussian Process Regression and Unscented Transform,“ in *2023 European Control Conference (ECC)*, IEEE, 2023, S. 1–8.
- [176] M. Hesse, L. Schwarzer, J. Timmermann und A. Trächtler, „Robust and Efficient Hybrid Optimal Control via Gaussian Process Regression and Multiple Shooting With Experimental Validation on a Double Pendulum on a Cart,“ *PAMM*, Jg. 25, Nr. 2, 2025.
- [177] J. Umlauft, T. Beckers, M. Kimmel und S. Hirche, „Feedback linearization using Gaussian processes,“ in *2017 IEEE 56th Annual Conference on Decision and Control (CDC)*, IEEE, 2017, S. 5249–5255.
- [178] G. G. Harman, *Wire bonding in microelectronics*, 3. Auflage. New York, NY: McGraw-Hill, 2010.
- [179] B. Shahriari, K. Swersky, Z. Wang, R. P. Adams und N. de Freitas, „Taking the Human Out of the Loop: A Review of Bayesian Optimization,“ *Proceedings of the IEEE*, Jg. 104, Nr. 1, S. 148–175, 2016.
- [180] M. Hesse, M. Hunstig, J. Timmermann und A. Trächtler, „Batch Constrained Bayesian Optimization for Ultrasonic Wire Bonding Feed-forward Control Design,“ in *Proceedings of the 11th International Conference on Pattern Recognition Applications and Methods*, SCITEPRESS - Science and Technology Publications, 2022, S. 383–394.
- [181] D. Luenberger, „An introduction to observers,“ *IEEE Transactions on Automatic Control*, Jg. 16, Nr. 6, S. 596–602, 1971.

- [182] D. Schröder, *Intelligente Verfahren: Identifikation und Regelung nichtlinearer Systeme*. Berlin und Heidelberg: Springer, 2010.
- [183] R. E. Kalman, „A New Approach to Linear Filtering and Prediction Problems,“ *Journal of Basic Engineering*, Jg. 82, Nr. 1, S. 35–45, 1960.
- [184] S. J. Julier und J. K. Uhlmann, „Unscented Filtering and Nonlinear Estimation,“ *Proceedings of the IEEE*, Jg. 92, Nr. 3, S. 401–422, 2004.
- [185] Z. Chen, C. Heckman, S. Julier und N. Ahmed, „Weak in the NEES?: Auto-tuning Kalman Filters with Bayesian Optimization,“ in *21st International Conference on Information Fusion*, IEEE, 2018, S. 1072–1079.
- [186] Z. Chen, N. Ahmed, S. Julier und C. Heckman, *Kalman Filter Tuning with Bayesian Optimization*, arXiv Preprint, arXiv:1912.08601, 2019.
- [187] A. Kullberg, I. Skog und G. Hendeby, „Online Joint State Inference and Learning of Partially Unknown State-Space Models,“ *IEEE Transactions on Signal Processing*, Jg. 69, S. 4149–4161, 2021.
- [188] J. Davila, L. Fridman und A. Poznyak, „Observation and identification of mechanical systems via second order sliding modes,“ *International Journal of Control*, Jg. 79, Nr. 10, S. 1251–1262, 2006.
- [189] S. K. Spurgeon, „Sliding mode observers: a survey,“ *International Journal of Systems Science*, Jg. 39, Nr. 8, S. 751–764, 2008.
- [190] Y. Shtessel, C. Edwards, L. Fridman und A. Levant, *Sliding Mode Control and Observation*. New York und Heidelberg: Birkhäuser, 2014.
- [191] A. T. Nelson, „Nonlinear estimation and modeling of noisy time-series by dual Kalman filtering methods,“ Dissertation, Oregon Graduate Institute of Science and Technology, 2000.
- [192] R. van der Merwe und E. A. Wan, „The square-root unscented Kalman filter for state and parameter-estimation,“ in *Proceedings of 2001 IEEE International Conference on Acoustics, Speech, and Signal Processing*, Piscataway, NJ: IEEE Operations Center, 2001, S. 3461–3464.
- [193] R. van der Merwe, „Sigma-Point Kalman Filters for Probabilistic Inference in Dynamic State-Space Models,“ Dissertation, Oregon Health & Science University, 2004.
- [194] C. Schweers, „Adaptive Sigma-Punkte-Filter-Auslegung zur Zustands- und Parameterschätzung an Black-Box-Modellen,“ Dissertation, Universität Paderborn, Paderborn, 2017.
- [195] G. Kutyniok, *Theory and Applications of Compressed Sensing*, arXiv Preprint, arXiv:1203.3815, 2012.
- [196] B. P. Gibbs, *Advanced Kalman Filtering, Least-Squares and Modeling: A Practical Handbook*. Hoboken, NJ: John Wiley & Sons, 2011.
- [197] R.-S. Götte und J. Timmermann, „Approximating a Laplacian Prior for Joint State and Model Estimation within an UKF,“ *IFAC-PapersOnLine*, Jg. 56, Nr. 2, S. 869–874, 2023.

- [198] R.-S. Götte und J. Timmermann, „Estimating States and Model Uncertainties Jointly by a Sparsity Promoting UKF,“ *IFAC-PapersOnLine*, Jg. 56, Nr. 1, S. 85–90, 2023.
- [199] D. L. Donoho, „Compressed Sensing,“ *IEEE Transactions on Information Theory*, Jg. 52, Nr. 4, S. 1289–1306, 2006.
- [200] E. J. Candès, J. Romberg und T. Tao, „Robust Uncertainty Principles: Exact Signal Reconstruction From Highly Incomplete Frequency Information,“ *IEEE Transactions on Information Theory*, Jg. 52, Nr. 2, S. 489–509, 2006.
- [201] A. Carmi, P. Gurfil und D. Kanevsky, „Methods for Sparse Signal Recovery Using Kalman Filtering With Embedded Pseudo-Measurement Norms and Quasi-Norms,“ *IEEE Transactions on Signal Processing*, Jg. 58, Nr. 4, S. 2405–2409, 2010.
- [202] S. J. Julier und J. J. LaViola, „On Kalman Filtering With Nonlinear Equality Constraints,“ *IEEE Transactions on Signal Processing*, Jg. 55, Nr. 6, S. 2774–2784, 2007.
- [203] S. M. Hirsh, D. A. Barajas-Solano und J. N. Kutz, „Sparsifying priors for Bayesian uncertainty quantification in model discovery,“ *Royal Society open science*, Jg. 9, Nr. 2, 2022.
- [204] J. Piironen und A. Vehtari, „Sparsity information and regularization in the horseshoe and other shrinkage priors,“ *Electronic Journal of Statistics*, Jg. 11, Nr. 2, S. 5018–5051, 2017.
- [205] A. Bhadra, J. Datta, N. G. Polson und B. Willard, „Lasso Meets Horseshoe: A Survey,“ *Statistical Science*, Jg. 34, Nr. 3, S. 405–427, 2019.
- [206] R.-S. Götte und J. Timmermann, „Online Learning With Joint State and Model Estimation,“ *PAMM*, Jg. 25, Nr. 1, 2025.
- [207] B. Ritter, „Nonlinear State Estimation and Noise Adaptive Kalman Filter Design for Wind Turbines,“ Dissertation, Technische Universität Darmstadt, Darmstadt, 2020.
- [208] T. Schmitt und B. Ritter, „Data-based Identifiability and Observability Assessment for Nonlinear Control Systems using the Profile Likelihood Method,“ *IFAC-PapersOnLine*, Jg. 53, Nr. 2, S. 1114–1119, 2020.
- [209] R.-S. Götte, J. N. Klusmann und J. Timmermann, „Data-driven identification of disturbances using a sliding mode observer,“ in *Proceedings - 33. Workshop Computational Intelligence: Berlin, 23.-24. November 2023*, H. Schulte, F. Hoffmann und R. Mikut, Hrsg., Karlsruher Institut für Technologie (KIT), 2023, S. 113–123.
- [210] K. Pearson, „On lines and planes of closest fit to systems of points in space,“ *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, Jg. 2, Nr. 11, S. 559–572, 1901.
- [211] H. Hotelling, „Analysis of a complex of statistical variables into principal components,“ *Journal of Educational Psychology*, Jg. 24, Nr. 6, S. 417–441, 1933.

- [212] P. Sanguansat, *Principal Component Analysis: Engineering Applications*. Rijeka: IntechOpen, 2012.
- [213] A. Surana und A. Banaszuk, „Linear observer synthesis for nonlinear systems using Koopman Operator framework,“ *IFAC-PapersOnLine*, Jg. 49, Nr. 18, S. 716–723, 2016.
- [214] A. Surana, „Koopman operator based observer synthesis for control-affine nonlinear systems,“ in *IEEE 55th Conference on Decision and Control (CDC)*, Piscataway, NJ, 2016, S. 6492–6499.
- [215] A. Surana, M. O. Williams, M. Morari und A. Banaszuk, „Koopman operator framework for constrained state estimation,“ in *2017 IEEE 56th Annual Conference on Decision and Control (CDC)*, Piscataway, NJ, 2017, S. 94–101.
- [216] G. V. Iungo, C. Santoni-Ortiz, M. Abkar, F. Porté-Agel, M. A. Rotea und S. Leonardi, „Data-driven Reduced Order Model for prediction of wind turbine wakes,“ *Journal of Physics: Conference Series*, Jg. 625, 2015.
- [217] M. Netto und L. Mili, „A Robust Data-Driven Koopman Kalman Filter for Power Systems Dynamic State Estimation,“ *IEEE Transactions on Power Systems*, S. 1–10, 2018.
- [218] K. V. Mardia und R. J. Marshall, „Maximum Likelihood Estimation of Models for Residual Covariance in Spatial Regression,“ *Biometrika*, Jg. 71, Nr. 1, S. 135–146, 1984.
- [219] W. H. Syed, A. Eichler, A. Nawaz, B. Sharan und H. Werner, „Koopman-based Kalman Filter for Fault Detection for the Superconducting Radio Frequency Cavities of the European XFEL,“ in *2021 60th IEEE Conference on Decision and Control (CDC)*, 2021, S. 6855–6860.
- [220] L. Zeng, S. Sadati und C. Bergeles, „Koopman Operator-based Extended Kalman Filter for Cosserat Rod Wrench Estimation,“ in *2023 International Symposium on Medical Robotics (ISMR)*, 2023, S. 1–7.
- [221] K. Ogata, *Modern control engineering*. Upper Saddle River, NJ: Prentice Hall, 2002.
- [222] RoboGolfPro, *Robo GolfPro*, 2017.
- [223] N. Betzler, „The Effect of Differing Shaft Dynamics on the Biomechanics of the Golf Swing,“ Dissertation, Edinburgh Napier University, Edinburgh, 2010.
- [224] C. Cogswell, R. Hathaway und J. Voinovich, „Golf Putter Testing Mechanism,“ Senior Project Final Report, California Polytechnic State University, San Luis Obispo, 2016.
- [225] Golf Tips Magazine, *Become A Swing Machine*, Golf Tips Magazine, Hrsg., 2003.
- [226] The Irish Times, *The last swing of Iron Byron*, 1998.
- [227] Miyamae Co., *Miya ROBO V: Swing Robot with Rotating body, ROBO 10 from Miyamae*.

- [228] Miyamae Co., *Miya ROBO-10: Swing Robot with Rotating body, ROBO 10 from Miyamae*.
- [229] Miyamae Co., *Miya ROBO-3DX: Club Testing and Club Destruction Testing specific Golf ROBO-3DX*.
- [230] The Golf Wire, *ROB-OT Makes An Appearance At The America's Warrior Partnership Golf Tournament In La Quinta, CA*, 2018.
- [231] S. Karaman und E. Frazzoli, „Incremental Sampling-based Algorithms for Optimal Motion Planning,“ in *Robotics: Science and Systems VI*, Robotics: Science and Systems Foundation, 2010.
- [232] S. Karaman und E. Frazzoli, „Sampling-based algorithms for optimal motion planning,“ *The International Journal of Robotics Research*, Jg. 30, Nr. 7, S. 846–894, 2011.
- [233] Beckhoff, *Original-Betriebsanleitung AM8000 und AM8500: Synchron Servomotoren*, 2023.
- [234] N. Fittkau, „Detaillierte Modellbildung für den Golfroboter zur Verbesserung der Zustandsregelung und -beobachtung,“ Bachelorarbeit (Wirtschaftsingenieurwesen Maschinenbau), Universität Paderborn, Paderborn, 2020.
- [235] A. Junker, N. Fittkau, J. Timmermann und A. Trächtler, „Autonomous Golf Putting with Data-Driven and Physics-Based Methods,“ *Sixth IEEE International Conference on Robotic Computing (IRC)*, S. 134–141, 2022.
- [236] S. Ren, K. He, R. Girshick und J. Sun, *Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks*, arXiv Preprint, arXiv:1506.01497, 2015.
- [237] Transportation Research Board, *Tires and Passenger Vehicle Fuel Economy: Informing Consumers, Improving Performance – Special Report 286*. Washington, DC: The National Academies Press, 2006.
- [238] Jonas Sülwold, „Weiterentwicklung und Validierung des Golfroboters für Szenarien mit einer hügeligen Spielfläche,“ Masterarbeit, Universität Paderborn, Paderborn, 2024.
- [239] Barry Plummer, *Amateur Golfers Make Less Than 40 Percent Of Putts From This Crucial Length'... Arccos Data Reveals Stark Putting Truths*, Golfmonthly, Hrsg., 2024.
- [240] ETH Zürich, Institute for Dynamic Systems and Control, *Cubli*, 2016.
- [241] M. Gajamohan, M. Muehlebach, T. Widmer und R. D'Andrea, „The Cubli: A reaction wheel based 3D inverted pendulum,“ in *2013 European Control Conference*, 2013, S. 268–274.





## **Das Heinz Nixdorf Institut – Interdisziplinäres Forschungszentrum für Informatik und Technik**

Das Heinz Nixdorf Institut ist ein Forschungszentrum der Universität Paderborn. Es entstand 1987 aus der Initiative und mit Förderung von Heinz Nixdorf. Damit wollte er Ingenieurwissenschaften und Informatik zusammenführen, um wesentliche Impulse für neue Produkte und Dienstleistungen zu erzeugen. Dies schließt auch die Wechselwirkungen mit dem gesellschaftlichen Umfeld ein.

Die Forschungsarbeit orientiert sich an dem Programm „Dynamik, Vernetzung, Autonomie: Neue Methoden und Technologien für die intelligenten technischen Systeme von morgen“. In der Lehre engagiert sich das Heinz Nixdorf Institut in Studiengängen der Informatik, der Ingenieurwissenschaften und der Wirtschaftswissenschaften.

Heute wirken am Heinz Nixdorf Institut acht Professoren/in mit insgesamt 120 Mitarbeiterinnen und Mitarbeitern. Pro Jahr promovieren hier etwa 15 Nachwuchswissenschaftlerinnen und Nachwuchswissenschaftler.

## **Heinz Nixdorf Institute – Interdisciplinary Research Centre for Computer Science and Technology**

The Heinz Nixdorf Institute is a research centre within the University of Paderborn. It was founded in 1987 initiated and supported by Heinz Nixdorf. By doing so he wanted to create a symbiosis of computer science and engineering in order to provide critical impetus for new products and services. This includes interactions with the social environment.

Our research is aligned with the program “Dynamics, Networking, Autonomy: New methods and technologies for intelligent technical systems of tomorrow”. In training and education the Heinz Nixdorf Institute is involved in many programs of study at the University of Paderborn. The superior goal in education and training is to communicate competencies that are critical in tomorrows economy.

Today eight Professors and 120 researchers work at the Heinz Nixdorf Institute. Per year approximately 15 young researchers receive a doctorate.




## Zuletzt erschienene Bände der Verlagsschriftenreihe des Heinz Nixdorf Instituts

- Bd. 406 OLEFF, C.: Proaktives Management von Anforderungsänderungen in der Entwicklung komplexer technischer Systeme. Dissertation, Fakultät für Maschinenbau, Universität Paderborn, Verlagsschriftenreihe des Heinz Nixdorf Instituts, Band 406, Paderborn, 2022 – ISBN 978-3-947647-25-5
- Bd. 407 JAVED, A. R.: Mixed-Signal Baseband Circuit Design for High Data Rate Wireless Communication in Bulk CMOS and SiGe BiCMOS Technologies. Dissertation, Fakultät für Elektrotechnik, Informatik und Mathematik, Universität Paderborn, Verlagsschriftenreihe des Heinz Nixdorf Instituts, Band 407, Paderborn, 2022 – ISBN 978-3-947647-26-2
- Bd. 408 DUMITRESCU, R. KOLDEWEY, C.: Daten-gestützte Projektplanung. Fachbuch. Fakultät für Elektrotechnik, Informatik und Mathematik, Universität Paderborn, Verlagsschriftenreihe des Heinz Nixdorf Instituts, Band 408, Paderborn, 2022 – ISBN 978-3-947647-27-9
- Bd. 409 PÖHLER, A.: Automatisierte dezentrale Produktionssteuerung für cyber-physische Produktionssysteme mit digitaler Repräsentation der Beschäftigten. Dissertation, Fakultät für Maschinenbau, Universität Paderborn, Verlagsschriftenreihe des Heinz Nixdorf Instituts, Band 409, Paderborn, 2022 – ISBN 978-3-947647-28-6
- Bd. 410 RÜDDENKLAU, N.: Hardware-in-the-Loop-Simulation von HD-Scheinwerfer-Steuergeräten zur Entwicklung von Lichtfunktionen in virtuellen Nachtfahrten. Dissertation, Fakultät für Maschinenbau, Universität Paderborn, Verlagsschriftenreihe des Heinz Nixdorf Instituts, Band 410, Paderborn, 2023 – ISBN 978-3-947647-29-3
- Bd. 411 BIEMELT, P.: Entwurf und Analyse modell-prädiktiver Regelungsansätze zur Steigerung des Immersionsempfindens in inter-aktiven Fahrsimulationen. Dissertation, Fakultät für Maschinenbau, Universität Paderborn, Verlagsschriftenreihe des Heinz Nixdorf Instituts, Band 411, Paderborn, 2023 – ISBN 978-3-947647-30-9
- Bd. 412 HAAKE, C.-J., MEYER AUF DER HEIDE, F., PLATZNER, M., WACHSMUTH, H., WEHRHEIM, H. (Eds.): On-The-Fly Computing - Individualized IT-Services in dynamic markets, Collaborative Research Centre 901 (2011 - 2023), Verlagsschriftenreihe des Heinz Nixdorf Instituts, Band 412, Paderborn, 2023 – ISBN 978-3-947647-31-6
- Bd. 413 DUMITRESCU, R.; HÖLZLE, K. (Hrsg.): Vorausschau und Technologieplanung. 17. Symposium für Vorausschau und Technologieplanung, Heinz Nixdorf Institut, 14./15. September 2023, Berlin-Brandenburgische Akademie der Wissenschaften, Berlin, Verlagsschriftenreihe des Heinz Nixdorf Instituts, Band 413, Paderborn, 2023 – ISBN 978-3-947647-32-3
- Bd. 414 ABUGHANNAM, S.: Low-power Direct-detection Wake-up Receiver at 2.44 GHz for Wireless Sensor Networks. Dissertation, Fakultät für Elektrotechnik, Informatik und Mathematik, Universität Paderborn, Verlagsschriftenreihe des Heinz Nixdorf Instituts, Band 414, Paderborn, 2024 – ISBN 978-3-947647-33-0
- Bd. 415 REINHOLD, J.: Systematik zur muster-basierten Transformation von Wertschöpfungssystemen für Smart Services. Dissertation, Fakultät für Maschinenbau, Universität Paderborn, Verlagsschriftenreihe des Heinz Nixdorf Instituts, Band 415, Paderborn, 2024 – ISBN 978-3-947647-34-7
- Bd. 416 YANG, X.: Eine Methode zur Unterstüt-zung von Entscheidungen bei der Ent-wicklung modularer Leichtbauprodukte. Dissertation, Fakultät für Maschinenbau, Universität Paderborn, Verlagsschriftenreihe des Heinz Nixdorf Instituts, Band 416, Paderborn, 2024 – ISBN 978-3-947647-35-4
- Bd. 417 GRÄLER, M.: Entwicklung adaptiver Ein-richtassistenzsysteme für Produktions-prozesse. Dissertation, Fakultät für Ma-schinenbau, Universität Paderborn, Ver-lagsschriftenreihe des Heinz Nixdorf Insti-tuts, Band 417, Paderborn, 2024 – ISBN 978-3-947647-36-1
- Bd. 418 RÖSMANN, D.: Menschenzentrierte Mon-tageplanung und -steuerung durch fähig-keitsorientierte Aufgabenzuordnung. Dis-sertation, Fakultät für Maschinenbau, Universität Paderborn, Verlagsschriften-reihe des Heinz Nixdorf Instituts, Band 418, Paderborn, 2024 – ISBN 978-3-947647-37-8

## Zuletzt erschienene Bände der Verlagsschriftenreihe des Heinz Nixdorf Instituts

- Bd. 419 BAHMANIAN, M.: Optoelectronic Phase-Locked Loop, Theory and Implementation. Dissertation, Fakultät für Elektrotechnik, Informatik und Mathematik, Universität Paderborn, Verlagsschriftenreihe des Heinz Nixdorf Instituts, Band 419, Paderborn, 2024 – ISBN 978-3-947647-38-5
- Bd. 420 HEIHOFF-SCHWEDE, J.: Spezifikations-technik zur Analyse, Gestaltung und Bewertung von Engineering-IT-Architekturen. Dissertation, Fakultät für Elektrotechnik, Informatik und Mathematik, Universität Paderborn, Verlagsschriftenreihe des Heinz Nixdorf Instituts, Band 420, Paderborn, 2024 – ISBN 978-3-947647-39-2
- Bd. 421 MEYER, M.: Systematik zur Planung und Verwertung von Betriebsdaten-Analysen in der strategischen Produktplanung. Dissertation, Fakultät für Elektrotechnik, Informatik und Mathematik, Universität Paderborn, Verlagsschriftenreihe des Heinz Nixdorf Instituts, Band 421, Paderborn, 2024 – ISBN 978-3-947647-40-8
- Bd. 422 MALENA, K.: Konzipierung, Analyse und Realumsetzung eines mehrstufigen modellprädiktiven Lichtsignalanlagenregelungssystems. Dissertation, Fakultät für Maschinenbau, Universität Paderborn, Verlagsschriftenreihe des Heinz Nixdorf Instituts, Band 422, Paderborn, 2024 – ISBN 978-3-947647-41-5
- Bd. 423 GÖTTE, R.-S.: Online-Schätzung von Modellungenauigkeiten zur automatischen Modelladaption unter Beibehaltung einer physikalisch-technischen Interpretierbarkeit. Dissertation, Fakultät für Maschinenbau, Universität Paderborn, Verlagsschriftenreihe des Heinz Nixdorf Instituts, Band 423, Paderborn, 2024 – ISBN 978-3-947647-42-2
- Bd. 424 PIERENKEMPER, C.: Systematik zur Entwicklung Leistungsstufen-basierter Industrie 4.0-Strategien. Dissertation, Fakultät für Elektrotechnik, Informatik und Mathematik, Universität Paderborn, Verlagsschriftenreihe des Heinz Nixdorf Instituts, Band 424, Paderborn, 2024 – ISBN 978-3-947647-43-9
- Bd. 425 DUMITRESCU, R.; HÖLZLE, K. (Hrsg.): Vorausschau und Technologieplanung. 18. Symposium für Vorausschau und Technologieplanung, Heinz Nixdorf Institut, 5./6. Dezember 2024, Berlin-Brandenburgische Akademie der Wissenschaften, Berlin, Verlagsschriftenreihe des Heinz Nixdorf Instituts, Band 425, Paderborn, 2024 – ISBN 978-3-947647-44-6
- Bd. 426 HESSE, M.: Interaktive Inbetriebnahme von Steuerungen und Regelungen für partiell bekannte dynamische Systeme mittels Gauß-Prozess-Regression. Dissertation, Fakultät für Maschinenbau, Universität Paderborn, Verlagsschriftenreihe des Heinz Nixdorf Instituts, Band 426, Paderborn, 2024 – ISBN 978-3-947647-45-3
- Bd. 427 PREUß, D.: Methodik zur automatisierten Extrahierung, Formalisierung und Handhabung von Änderungen technischer Anforderungen. Dissertation, Fakultät für Maschinenbau, Universität Paderborn, Verlagsschriftenreihe des Heinz Nixdorf Instituts, Band 427, Paderborn, 2024 – ISBN 978-3-947647-46-0
- Bd. 428 JUNKER, A.: Datengetriebene Modellbildung für nicht lineare mechatronische Systeme in regelungstechnisch verwertbarer Form. Dissertation, Fakultät für Maschinenbau, Universität Paderborn, Verlagsschriftenreihe des Heinz Nixdorf Instituts, Band 428, Paderborn, 2024 – ISBN 978-3-947647-47-7
- Bd. 429 THIELE, H.: Technologievorausschau auf Basis mathematischer Modellierung. Dissertation, Fakultät für Maschinenbau, Universität Paderborn, Verlagsschriftenreihe des Heinz Nixdorf Instituts, Band 429, Paderborn, 2025 – ISBN 978-3-947647-48-4





Im Zuge der Digitalisierung erfahren maschinelles Lernen und datengetriebene Methoden derzeit eine große Aufmerksamkeit in Wissenschaft und Industrie. Es fehlt jedoch an Grundlagenwissen und Verständnis, wie die datengetriebenen Methoden der Informatik mit bewährten modellbasierten Ingenieursmethoden wie dem modellbasierten Entwurf in der Mechatronik und Methoden der Regelungstechnik sinnvoll kombiniert werden können, um hybride Modelle zu erhalten. Diese ingenieurwissenschaftlichen Methoden basieren auf physikalischen Verhaltensmodellen, die eine besonders verdichtete und interpretierbare Darstellung von Wissen darstellen und insbesondere kausale Zusammenhänge beschreiben. Für spezifische regelungstechnische Anwendungen gibt es umfangreiches Vorwissen in Form von bekannten Strukturen und Informationen, wie z.B. (Teil-)Modelle oder Parametersätze, die auch bei der Anwendung von Methoden wie dem maschinellen Lernen genutzt werden sollten. Eine solche sinnvolle systematische Verknüpfung ist wissenschaftlich, insbesondere im Hinblick auf die industrielle Anwendung, noch nicht ausreichend untersucht worden und sehr vielversprechend. In diesem Beitrag werden die Ergebnisse der Nachwuchsforschungsgruppe DART – Datengetriebene Methoden in der Regelungstechnik vorgestellt. Das Hauptziel war es, die synergetische Kombination von modell- und datengetriebenen Methoden für regelungstechnische Aufgaben zu erforschen und es werden alle wichtigen Forschungsergebnisse aber auch die verwendeten Grundprinzipien des maschinellen Lernens in diesem Beitrag dargestellt.