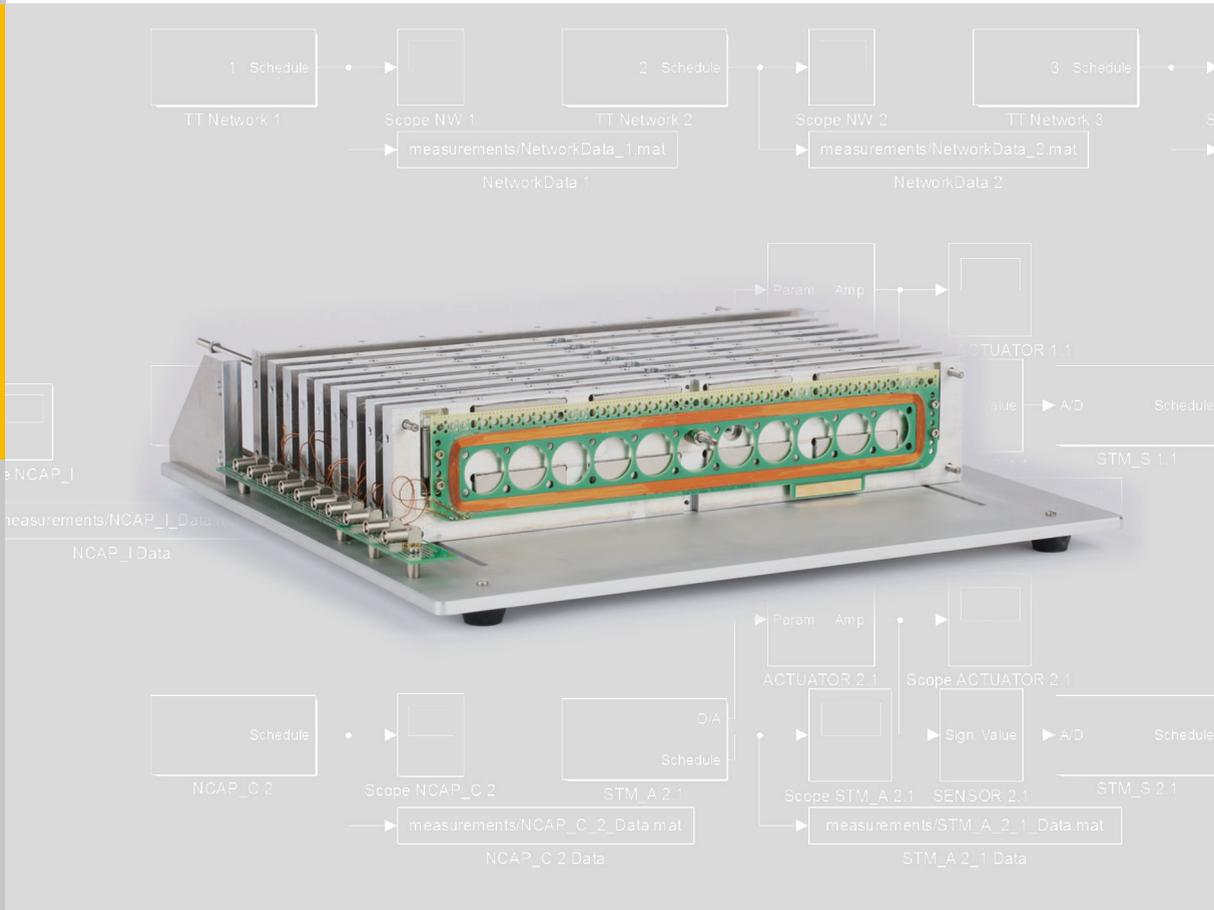


# Modellbasierte Ansteuerung räumlich ausgedehnter Aktuator- und Sensornetzwerke in der Strömungsregelung

Marcel Dück



Energie & Umwelt/  
Energy & Environment  
Band/ Volume 349  
ISBN 978-3-95806-193-4





Forschungszentrum Jülich GmbH  
Zentralinstitut für Engineering, Elektronik und Analytik (ZEA)  
Systeme der Elektronik (ZEA-2)

# Modellbasierte Ansteuerung räumlich ausgedehnter Aktuator- und Sensornetzwerke in der Strömungsregelung

Marcel Dück

Schriften des Forschungszentrums Jülich  
Reihe Energie & Umwelt / Energy & Environment

Band / Volume 349

---

ISSN 1866-1793

ISBN 978-3-95806-193-4

Bibliografische Information der Deutschen Nationalbibliothek.  
Die Deutsche Nationalbibliothek verzeichnet diese Publikation in der  
Deutschen Nationalbibliografie; detaillierte Bibliografische Daten  
sind im Internet über <http://dnb.d-nb.de> abrufbar.

Herausgeber und Vertrieb: Forschungszentrum Jülich GmbH  
Zentralbibliothek, Verlag  
52425 Jülich  
Tel.: +49 2461 61-5368  
Fax: +49 2461 61-6103  
E-Mail: [zb-publikation@fz-juelich.de](mailto:zb-publikation@fz-juelich.de)  
[www.fz-juelich.de/zb](http://www.fz-juelich.de/zb)

Umschlaggestaltung: Grafische Medien, Forschungszentrum Jülich GmbH

Druck: Grafische Medien, Forschungszentrum Jülich GmbH

Copyright: Forschungszentrum Jülich 2016

Schriften des Forschungszentrums Jülich  
Reihe Energie & Umwelt / Energy & Environment, Band / Volume 349

D 82 (Diss. RWTH Aachen University, 2016)

ISSN 1866-1793  
ISBN 978-3-95806-193-4

Vollständig frei verfügbar über das Publikationsportal des Forschungszentrums Jülich (JuSER)  
unter [www.fz-juelich.de/zb/openaccess](http://www.fz-juelich.de/zb/openaccess).



This is an Open Access publication distributed under the terms of the [Creative Commons Attribution License 4.0](https://creativecommons.org/licenses/by/4.0/),  
which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

# Vorwort

Diese Dissertation entstand während meiner Tätigkeit als Doktorand im Zentralinstitut für Engineering, Elektronik und Analytik, ZEA-2: Systeme der Elektronik in der Forschungszentrum Jülich GmbH.

Sie wurde im Rahmen der Forschergruppe „FOR1779: Aktive Widerstandsreduktion durch wellenförmige Oberflächenszillation“ durchgeführt. Der Deutschen Forschungsgemeinschaft danke ich in diesem Zusammenhang für die finanzielle Förderung dieses Projekts und allen Beteiligten der Forschergruppe für die gute Zusammenarbeit.

Für die Betreuung meiner Arbeit danke ich dem Leiter des Instituts für Regelungstechnik der RWTH Aachen, Herrn Univ.-Prof. Dr.-Ing. Dirk Abel. Ich danke Herrn Dr. Michael Schiek für die Betreuung im ZEA-2 in Jülich, für seine stets passenden Bemerkungen sowie das mir entgegengebrachte Vertrauen.

Danken möchte ich auch Herrn Univ.-Prof. Dr.-Ing. Wolfgang Schröder für das meiner Arbeit entgegengebrachte Interesse und die Übernahme des Korreferats, Herrn Univ.-Prof. Dr.-Ing. Lorenz Singheiser für die Übernahme des Vorsitzes der Prüfungskommission und Herrn Prof. Dr.-Ing. Heinrich Rake für seine Bereitschaft, als Beisitzer mitzuwirken.

Weiterhin danke ich Prof. Dr.-Ing Stefan van Waasen und allen Mitarbeitern des ZEA-2 für die Zusammenarbeit und die freundschaftliche Arbeitsatmosphäre. Namentlich hervorheben möchte ich Mario Schlösser, Roger Heil und Wolfgang Silex für ihren unermüdlichen Einsatz. Ein großer Dank gilt auch Lilesha Mulpuri, Sebastian Völkel und Florian Seidler und den Doktoranden des ZEA-2 für die angeregten Diskussionen.

Ich danke Herrn Prof. Dr. Eugênio de Bona Castelan Neto für seine Anregungen während meines Aufenthalts in der Universidade Federal de Santa Catarina in Florianópolis in Brasilien.

---

Meinen Eltern und meiner Schwester danke ich für die fortwährende Unterstützung und dass Sie mir meinen bisherigen Weg ermöglicht haben. Meiner Freundin Henrieke danke ich für ihren liebevollen Beistand.

Aachen, im November 2016

Marcel Dück

# Inhaltsangabe

Das Thema der vorliegenden Dissertation befasst sich mit der modellbasierten Entwicklung eines echtzeitfähigen, räumlich ausgedehnten Aktuator- und Sensornetzwerkes für den Einsatz in der Strömungsregelung. Innerhalb eines kaskadierten Regelkreises wird die externe Strömungsregelung über das Netzwerk mit definierten Schnittstellen als Model-in-the-loop an das elektromagnetische Aktuator-System zur Strömungsbeeinflussung angebunden. Die Strömungsbeeinflussung wird mithilfe von transversalen Oberflächenwellen auf einer drei Millimeter starken Aluminiumplatte realisiert.

Dieses Vorgehen ermöglicht sowohl die Durchführung von Experimenten zur Widerstandsbeeinflussung im Windkanal, als auch die Analyse der Unterschiede zwischen den Netzwerkkonfigurationen, die zur Bestimmung einer günstigen Topologie und Aufgabenverteilung führen. Dies bildet die Grundlage zur Spezifikation der Netzwerkkonfigurationen für die technische Umsetzung eines räumlich ausgedehnten Aktuator- und Sensornetzwerkes.

Innerhalb des Netzwerkes werden die notwendigen Aufgaben anhand eines Modells auf die entsprechenden Netzwerkknoten abgebildet. Zur Kommunikation werden die Kommunikationsschichten entsprechend des OSI-Referenzmodells definiert. Auf der Transportschicht wird ein echtzeitfähiges Protokoll integriert und anhand einer Simulation verifiziert. Verschiedene Netzwerksimulationen werden hinsichtlich unterschiedlicher Randbedingungen und Konfigurationen untersucht und die Ergebnisse diskutiert.

Zur Ansteuerung des elektromagnetischen Aktuator-Systems wird ein Verfahren zur Berechnung und echtzeitigen Umsetzung glatter Signalübergänge zwischen unterschiedlich parametrisierten sinusförmigen Signalen vorgestellt. Mithilfe der Wellenregelung wird das System stabilisiert und die angestrebte Genauigkeit der Wellenbewegung sichergestellt. Dazu wird eine angepasste modellbasierte Iterativ Lernende Regelung mit variablen Verstärkungsfaktoren entwickelt.



# Abstract

The topic of this thesis deals with the model-based development of a real-time, spatially enlarged actuator and sensor network for use in flow control. Within a cascaded control loop, the external flow control is connected via the network with defined interfaces as model-in-the-loop to the electromagnetic actuator system for influencing the flow. The flow is influenced by means of transversal surface waves on a three millimeter thick aluminum plate.

This approach allows both experiments in the wind tunnel as well as the analysis of differences in network configurations, which lead to the determination of a favorable topology and computation distribution. This forms the basis for the specification of network configurations for the technical implementation of a spatially enlarged actuator and sensor network.

The necessary tasks are mapped to the corresponding network nodes using a model. The communication layers are defined according to the OSI reference model. A real-time protocol is integrated on the transport layer and verified by a simulation. Various network simulations are investigated with regard to different boundary conditions and configurations and the results are discussed.

A method for real-time calculation and application of smooth signal transitions between differently parameterized sinusoidal signals for driving the electromagnetic actuator system is presented. Using the wave control, the system is stabilized and the accuracy of the wave motion is ensured. For this purpose, an adapted model-based iterative learning control with gain switching is developed.



# Inhaltsverzeichnis

<b>Abkürzungen und Akronyme</b>	<b>XIII</b>
<b>1 Einleitung</b>	<b>1</b>
1.1 Konzepte zur passiven und aktiven Strömungsbeeinflussung . . .	2
1.2 Netzwerke . . . . .	6
1.3 Ziel und Gliederung dieser Arbeit . . . . .	7
<b>2 Grundlagen: Aktuator- und Sensornetzwerke</b>	<b>11</b>
2.1 Grundlegender Aufbau von Netzwerken . . . . .	11
2.1.1 Das OSI-Referenzmodell . . . . .	13
2.1.2 Das Internet-Referenzmodell . . . . .	14
2.1.3 Echtzeitfähige Protokolle . . . . .	16
2.1.4 Protokolle der Anwendungsschicht . . . . .	17
2.2 Aktuator- und Sensornetzwerke . . . . .	19
2.2.1 Netzwerke in der Regelungstechnik . . . . .	19
2.2.2 Netzwerke in der industriellen Automatisierung und der Fahrzeugtechnik . . . . .	21
2.3 Netzwerkknoten . . . . .	22
2.4 Werkzeuge zur Modellierung und Ansteuerung . . . . .	23
<b>3 Netzwerkconcept des kaskadierten Regelkreises</b>	<b>27</b>
3.1 Spezifikation des Netzwerkes . . . . .	27
3.1.1 Allgemeine Anforderungen für großflächige Netzwerke .	28
3.1.2 Applikationsanforderungen . . . . .	30
3.2 Validierungsstrategie . . . . .	35
3.3 Netzwerkschichten gemäß OSI-Referenzmodell . . . . .	37
3.3.1 Anwendungsschicht . . . . .	37
3.3.2 Transportschicht . . . . .	39
3.3.3 Netzwerkschicht . . . . .	39
3.3.4 Externe Netzwerkschnittstellen . . . . .	41
3.4 Modellierung des Aktuator- und Sensornetzwerkes . . . . .	42
3.4.1 Werkzeuge und Implementierung . . . . .	43
3.4.2 Modellierung der Netzwerkschicht . . . . .	45

3.4.3	Modellierung der Transportschicht . . . . .	46
3.4.4	Modellierung der Anwendungsschicht . . . . .	53
3.4.5	Model-in-the-Loop-Implementierung . . . . .	58
3.5	Modellkonfiguration und Simulation . . . . .	60
3.6	Zusammenfassung . . . . .	62
<b>4</b>	<b>Erzeugung von Oberflächenwellen im aktuatornahen Regelkreis</b>	<b>65</b>
4.1	Elektromagnetisches Aktuator-System zur Erzeugung von Oberflächenwellen . . . . .	65
4.2	Echtzeitige Parametervariation für die Oberflächenwellen (Open Loop) . . . . .	68
4.2.1	Generierung glatter Signalübergänge . . . . .	71
4.3	Aktuator-Modell . . . . .	74
4.4	Regelungskonzept: Iterativ Lernende Regelung . . . . .	78
4.4.1	Grundlagen zur Iterativ Lernenden Regelung . . . . .	78
4.4.2	Konvergenz, Design und Implementierung . . . . .	79
4.4.3	Simulationsergebnisse . . . . .	85
4.5	Zusammenfassung . . . . .	86
<b>5</b>	<b>Modellvalidierung und Model-in-the-Loop-Simulation</b>	<b>89</b>
5.1	Mikrocontrollerbasiertes Testnetzwerk . . . . .	89
5.2	Validierung der Kommunikation . . . . .	91
5.2.1	Validierung der Netzwerkschicht . . . . .	92
5.2.2	Validierung der Transportschicht . . . . .	99
5.2.3	Validierung der Anwendungsschicht . . . . .	105
5.3	Validierung des Netzwerkmodells . . . . .	108
5.3.1	Evaluierung konkreter Netzwerkkonfigurationen . . . . .	109
5.3.2	Analytischer Ansatz zur Optimierung der Zykluszeiten . . . . .	115
5.3.3	Synchronisation . . . . .	118
5.3.4	Fehlerbehandlung . . . . .	120
5.3.5	Bewertung der Evaluation des Netzwerkes und der Mo- dellimplementierung . . . . .	124
5.4	Simulationsbasierte Validierung der Aktuator- und Sensornetz- werke . . . . .	124
5.4.1	Simulation des Netzwerkes für homogene Strömungsbe- dingungen . . . . .	126
5.4.2	Simulation des Netzwerkes für inhomogene Strömungs- bedingungen . . . . .	129
5.4.3	Bewertung der Simulationen . . . . .	132

5.5	Steuerung und Regelung des Aktuator-Systems . . . . .	132
5.5.1	Echtzeitige Generierung und Änderung der transversalen Oberflächenwellen . . . . .	132
5.5.2	Modellbasierte Entwicklung der Wellenregelung (innerer Regelkreis) . . . . .	135
5.6	Ankopplung an die Strömungsregelung (Model-in-the-Loop-Simulation) . . . . .	137
5.7	Zusammenfassung . . . . .	139
<b>6</b>	<b>Diskussion und Ausblick</b>	<b>141</b>
	<b>Literaturverzeichnis</b>	<b>143</b>



# Abkürzungen und Akronyme

AIA	Aerodynamisches Institut der RWTH Aachen
ARP	Address Resolution Protocol
ASCII	American Standard Code for Information Interchange
CPS	Cyber-Physische-Systeme
et al.	et alii
FTP	File Transfer Protocol
HTTP	Hypertext Transfer Protocol
IEC	International Electrotechnical Commission
IEEE	Institute of Electrical and Electronics Engineers
ILR	Iterativ Lernende Regelung
IoT	Internet of Things
ISO	International Organization for Standardization
IRT	Institut für Regelungstechnik der RWTH Aachen
LAN	Local Area Network
LLC	Logical Link Control
MAC	Medium Access Control
MTS	Master-Token-Slave
MTU	Maximum Transfer Unit
NCAP	Network Capable Application Processor
NIST	National Institute of Standards and Technology
NTP	Network Time Protocol
OSI	Open Systems Interconnection
PCB	Printed Circuit Board
PTP	Precision Time Protocol
RFC	Request For Comments
RTC	Real Time Clock
RTP	Real-Time Transport Protocol
SMTP	Simple Mail Transfer Protocol
STM	Smart Transducer Module
TCP	Transmission Control Protocol
UDP	User Datagram Protocol
XML	Extensible Markup Language



# 1 Einleitung

Heutige Transportmittel wie Flugzeuge, Züge und Schiffe sind ein wichtiger Bestandteil des Alltags. Das Verkehrsaufkommen steigt gerade in diesen Bereichen kontinuierlich an. Damit erhöhen sich Energiebedarf und Emissionen. Die Weiterentwicklung der Technik muss diese Effekte möglichst weitgehend kompensieren. Die Technologieentwicklung wird dazu in vielen Teilbereichen vorangetrieben, unter anderem durch die Verwendung leichter oder robusterer Materialien sowie Veränderungen an den Antriebskonfigurationen.

Ein großes Potential wird dabei der Verringerung des aerodynamischen Reibungswiderstandes zugeschrieben [52]. Hierfür werden sowohl passive als auch aktive Methoden diskutiert [46].

Im Rahmen der Forschergruppe FOR 1779 „Aktive Widerstandsreduktion durch wellenförmige Oberflächenoszillation“ werden robuste aktive Methoden für die Strömungsbeeinflussung mittels transversaler Oberflächenwellen entwickelt. Dieser Ansatz wird in der Anwendung die gesamte Oberfläche beispielsweise eines Flugzeugs mit einbeziehen und setzt daher weiträumig verteilte Aktuatoren und Sensoren voraus, welche für die Berücksichtigung zum Beispiel von instationären Strömungsbedingungen durch eine zentrale Regelung angesprochen beziehungsweise ausgelesen werden müssen. Die modellbasierte Entwicklung eines solchen echtzeitfähigen, räumlich ausgedehnten Aktuator- und Sensornetzwerkes für den Einsatz in der Strömungsregelung ist Thema der vorliegenden Arbeit. Bereits in der Entwicklungsphase wird das implementierte Netzwerkmodell als Model-in-the-Loop-Simulation für die Ansteuerung eines elektromagnetischen Aktuator-Systems zur geregelten Erzeugung der transversalen Oberflächenwellen auf einer Aluminiumoberfläche im Rahmen der Methodenentwicklung in Windkanalversuchen verwendet.

Nachfolgend werden in dieser Einleitung aktuelle Konzepte für die passive und aktive Strömungsbeeinflussung sowie der entsprechende wissenschaftliche Ansatz der Forschergruppe FOR 1779 vorgestellt. Nach einer kurzen Einführung in die Thematik der Netzwerke schließt dieses Kapitel mit einer Darstellung der Ziele und Gliederung dieser Arbeit.

# 1.1 Konzepte zur passiven und aktiven Strömungsbeeinflussung

Für die Beeinflussung von Strömungen werden unterschiedliche Konzepte eingesetzt. Diese lassen sich zunächst in zwei Klassen, die passiven und die aktiven Methoden, unterteilen. Karniadakis und Choi beschreiben in [46] die Untersuchung beider Ansätze sowohl mittels physikalischer Experimente als auch mit numerischen Simulationen. Gegenstand dieser Ansätze ist die Veränderung der Wandschubspannung. Mit dem Ziel der Widerstandsbeeinflussung werden die Auswirkungen von speziell beschaffenen Oberflächen oder Oberflächenbewegungen auf die turbulente Grenzschicht beobachtet und simuliert. Die experimentelle Evaluierung erfolgt dabei in der Regel in niedrigen Reynoldszahlbereichen.

Die sogenannten Riblets, also Mikrorillen, die auf einer glatten Fläche aufgebracht werden, sind eine vielfach untersuchte Methode zur passiven Strömungsbeeinflussung. Die Abmessungen der Riblets liegen in der Größenordnung der viskosen Unterschicht und haben üblicherweise einen dreieckigen oder halbkreisförmigen Querschnitt. Sie bewirken eine Reduktion des turbulenten Reibungswiderstandes von bis zu zehn Prozent [46].

Für den experimentellen Nachweis der Widerstandsreduktion durch aktive Beeinflussung der Strömung werden verschiedene Aktuator-Konzepte eingesetzt [15]. Dabei handelt es sich beispielsweise um fluidische Aktuatoren, die einen (zusätzlichen) Partikelstrom in die Strömung einbringen oder absaugen. Weiterhin werden Plasma-Aktuatoren verwendet, die das Fluid ionisieren und mithilfe einer Elektrode beschleunigen. Außerdem werden Aktuator Systeme eingesetzt, die auf direkter elektromagnetischer oder magnetohydrodynamischer Interaktion mit dem Fluid basieren. Die Aktuatoren zur Erzeugung wandernder Oberflächenwellen werden hauptsächlich piezo-elektrisch angetrieben. Die mögliche Auslenkung der Aktuatoren bei Frequenzen in Bereich von ein bis zwei Kilohertz liegt in der Regel zwischen zehn und  $100 \mu\text{m}$  [15].

Zur Messung und Beobachtung der Strömungen werden verschiedene Methoden eingesetzt. Dreidimensionale Strömungsfelder werden experimentell mit der *Particle Image Velocimetry* (PIV) bestimmt. Die *Micro-Particle-Tracking Velocimetry* (PTV) dient zur Messung des Geschwindigkeitsvektors individueller Partikel, die sich in der Strömung befinden. Weiterhin werden Hitzdrahtsensoren zur Bestimmung der Geschwindigkeit einer eindimensionalen Strömung verwendet. Schubspannungssensoren ermöglichen die Messung der

lokalen Wandschubspannung. Das Messprinzip beruht auf der Detektion der Verbiegung von Mikrozyllindern durch das Strömungsfeld [33].

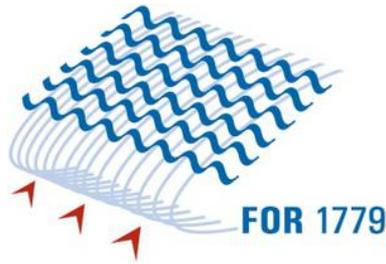
Die bisher untersuchten aktiven Methoden lassen sich in Bewegungsformen in der Ebene und senkrecht zur Wand unterscheiden. Die Bewegungen in der Ebene wurden bisher vorwiegend in turbulenten Rohr- und Kanalströmungen untersucht. Dabei wurde eine Widerstandsreduktion von bis zu 40% beobachtet. Zudem wurde unter bestimmten Voraussetzungen eine mögliche netto Energieeinsparung von bis zu 10% nachgewiesen [46].

Simulationen zeigen, dass mit transversalen Oberflächenoszillationen eine Widerstandsreduktion von bis zu 30% erzielt werden kann. Dabei werden die Wellen quer zur Strömungsrichtung senkrecht zur umströmten Oberfläche erzeugt. Der Betrag der Widerstandsreduktion hängt von der Frequenz, Länge und Amplitude der Wellen ab [46].

## FOR 1779

Die Forschergruppe FOR 1779 bearbeitet das Forschungsprojekt „Aktive Widerstandsreduktion durch wellenförmige Oberflächenoszillation“. Dies gehört zum Bereich der grundlagenorientierten Technologieforschung. Innerhalb des Projektes werden robuste Methoden für die Strömungsregelung mithilfe transversaler Oberflächenwellen entwickelt. Dabei sollen auch die Kombination dieser aktiven Methode mit dem passiven Ansatz der Riblet-strukturierten Oberfläche analysiert werden (siehe Abb. 1.1). Es werden Strömungen im Bereich der auf die Impulsverlustdicke  $\theta$  bezogene Reynoldszahl  $Re_\theta$  zwischen  $10^3$  und  $10^4$  untersucht. Für die Entwicklung der Methoden werden sowohl experimentelle Messungen im Windkanal als auch numerische Simulationen durchgeführt. Ziel ist das Verständnis der grundlegenden Mechanismen der Turbulenzbeeinflussung durch diese Methoden. Das übergeordnete Ziel ist die Reduktion des Energieverbrauchs und der Geräuschemission. Ein zentraler Aspekt ist die Verbesserung der Gesamtenergiebilanz, insofern ist auch die für die Strömungsbeeinflussung notwendige Energie Gegenstand der Forschung.

Vergleichbare Experimente zur Turbulenzbeeinflussung mit künstlichen Oberflächen und transversalen Oberflächenwellen liefern eine maximale relative Widerstandsreduktion von 13% [81]. Bei diesen Experimenten wurden die

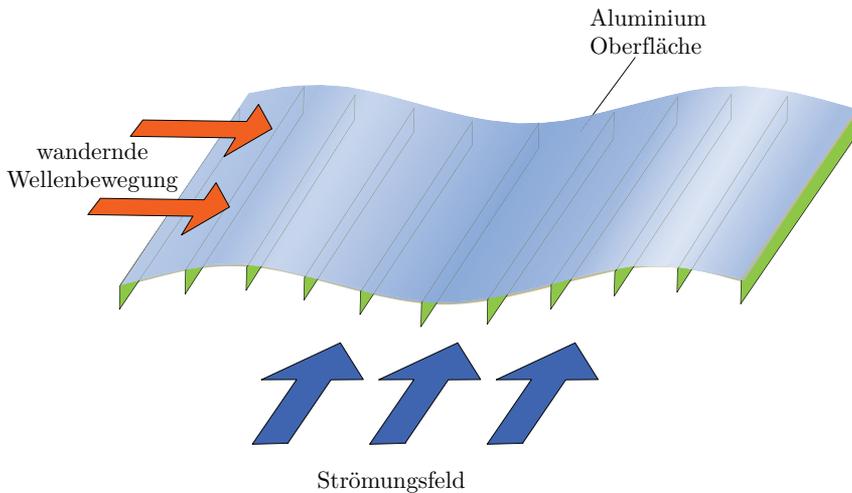


**Abb. 1.1:** Das Logo der Forschergruppe FOR 1779 zeigt eine schematische Darstellung des Projektinhalts. Dieser beinhaltet eine Kombination aus Riblets mit transversalen Oberflächenwellen zur Reduktion des turbulenten Reibungswiderstandes.

Oberflächenwellen auf speziellen, sehr dünnen und damit flexiblen Materialien erzeugt. Für die Außenhaut von großen Transportmitteln sind jedoch nicht beliebig flexible und beliebig dünne Materialien einsetzbar.

Innerhalb der Forschergruppe FOR 1779 liegt das Augenmerk daher auf der Verwendung von Aluminiumoberflächen für die Experimente im Windkanal. Die aktive Turbulenzbeeinflussung mit transversalen Oberflächenwellen quer zur Anströmungsrichtung wird über einer ebenen Aluminiumplatte untersucht (siehe Abb. 1.2). Das Material ist weniger flexibel und die Erzeugung wellenförmiger transversaler Bewegungen erfordert somit höhere Kräfte, als für die angesprochenen sehr dünnen und flexiblen Oberflächen notwendig wären. Dennoch müssen die Oberflächenwellen genau der Vorgabe entsprechen, da nur so die experimentellen Ergebnisse mit den numerischen Simulationen abgeglichen und hinsichtlich des Verständnisses der grundlegenden Mechanismen der Strömungsbeeinflussung interpretiert werden können. Hierfür muss die Oberflächenbewegung erfasst und als Eingangsgröße für die Regelung der Kraftübertragung der Aktuatoren auf die Oberfläche genutzt werden.

Für das Verständnis der Beeinflussung der turbulenten Grenzschicht werden sowohl numerische Simulationen (Large Eddy Simulationen (LES)) als auch experimentelle Untersuchungen in einem subsonischen Windkanal durchgeführt. Detaillierte Beschreibungen dazu finden sich in [49], [50] und [72]. Innerhalb der Windkanalexperimente werden verschiedene Arbeitspunkte in einer stationären und homogenen Anströmung untersucht. Dafür werden  $\mu$ -PTV- und PIV-Messungen sowie zeitlich und räumlich hochaufgelöste Hitzdrahtmessungen vor und hinter der aktuierten Oberfläche durchgeführt. Die



**Abb. 1.2:** Eine wandernde Wellenbewegung wird auf eine Aluminiumoberfläche aufgebracht. Diese beeinflusst das Strömungsfeld und ermöglicht eine aktive Widerstandsreduktion.

Kombination dieser Messmethoden ermöglicht die Vermessung der gesamten Grenzschicht, stromauf und stromab der aktuierten Oberfläche, sowie die Geschwindigkeitsmessung im wandnahen Bereich. In Experimenten mit einer Reynoldszahl von  $Re_\theta = 1200$  bezogen auf die Impulsverlustdicke  $\theta$  konnte bereits eine Widerstandsreduktion von 3,4% nachgewiesen werden [72]. Dafür wurde mit dem in dieser Arbeit beschriebenen elektromagnetischen Aktuator-System auf einer glatten 0,3 mm starken Aluminiumoberfläche eine wandernde sinusförmige Welle mit einer Amplitude von  $375 \mu\text{m}$ , einer Frequenz von 81 Hz sowie einer Wellenlänge von 160 mm erzeugt. Die Versuche wurden auch mit einer der Reynoldszahl angepassten Riblet-strukturierten Aluminiumoffläche der gleichen Stärke durchgeführt. Im unaktuierten Fall wurde eine Widerstandsreduktion von 4,7% gemessen. Für die aktuierte Ribletoberfläche wurden 9,4% ermittelt. Die Kombination von aktiven und passiven Methoden haben sich also gegenseitig verstärkt [52]. Der Vergleich von Simulation und Experiment zeigt eine gute Übereinstimmung und belegt, dass die Geschwindigkeitsmessungen unmittelbar stromab der aktuierten Oberfläche eine verlässliche Größe für die Bestimmung der relativen Widerstandsreduktion bezogen auf die unaktuierte Oberfläche liefern [55].

Die Simulationen und Experimente beziehen sich bisher auf feste Arbeits-

punkte, also stationäre und homogene Anströmungen. Bei wechselnden Anströmungsbedingungen müssen Störungen unterdrückt werden, um den mittleren Reibungswiderstand gering zu halten [35]. Die Simulationen und Windkanalexperimente mit einer oszillierenden Oberfläche ermöglichen die Ableitung der Auswirkungen verschiedener Parametersätze in unterschiedlichen Arbeitspunkten. Die gewonnenen Daten lassen sich dazu verwenden, modellbasierte Regelstrategien für den Einsatz in einer zentralen Regelung zu entwickeln. In [69] wird beschrieben, dass die Nichtlinearitäten zwischen den verschiedenen untersuchten Arbeitspunkten durch Interpolation ausgeglichen werden können, um die Voraussetzungen für die modellbasierte Strömungsregelung zu schaffen.

## 1.2 Netzwerke

Der Begriff „Netzwerk“ ist weit verbreitet und wird in verschiedenen Bereichen verwendet. Es existieren Soziale Netzwerke, Produktionsnetzwerke, Unternehmensnetzwerke oder auch Wissenschaftliche Netzwerke. In den Naturwissenschaften spricht man sowohl von biologischen als auch von künstlichen neuronalen Netzen. In dieser Arbeit wird der Begriff Netzwerk gemäß der Definition von Kersken im Sinne eines Computer- oder Rechnernetzes verwendet:

„Ein *Netzwerk* ist eine Verbindung mehrerer Computer zum Zweck des Datenaustauschs, für verteilte Anwendungen oder auch für die Kommunikation zwischen ihren Benutzern.“ (Kersken, [47], S. 537).

Die grundsätzliche Aufgabe eines Netzwerkes ist die Verbindung von sogenannten Netzwerkteilnehmern zum Informationsaustausch. Im „digitalen Zeitalter“ spielt die Vernetzung eine immer größere Rolle, dabei werden die Teilnehmer immer kompakter, leistungsstärker und energieeffizienter.

Für die Kommunikation in Netzwerken werden zwischen den Teilnehmern Regeln zur Kommunikation abgestimmt. Diese Vereinbarungen werden Protokolle genannt. Ohne standardisierte Protokolle würden keine digitalen Medien existieren und es gäbe keine der modernen Kommunikationskanäle wie E-Mail oder Kurzmitteilungsdienste. Assistenzsysteme im Fahrzeug wären genauso wie der Onlinehandel nicht vorstellbar.

Ein aktuelles Beispiel zur Kommunikation und Verarbeitung von Sensordaten durch kompakte und leistungsstarke Geräte ist allgegenwärtig: Die Entwicklung des Smartphones ermöglicht die Verwendung komplexer Anwendungen

zur ständigen Übermittlung beispielsweise von Positionsdaten über eine permanente Internetverbindung. Die langzeitige Funktionalität wird mithilfe eines kleinen Akkus sichergestellt. Als Übertragungsmedium zur Vernetzung dient das Mobilfunknetz oder Wi-Fi.

Neben der Verwendung von Rechnernetzen im privaten Umfeld spielen diese in der industriellen Automatisierung und auch im Bereich der Kraftfahrzeugtechnik eine große Rolle. Die Teilnehmer dieser Netzwerke sind in der Regel eingebettete Systeme auf der Basis von Mikrocontrollern. In Netzwerken für die Umsetzung von Regelungsaufgaben überwiegen in der Regel die Sensoren gegenüber den Aktuatoren. Insofern hat sich für diese Netzwerke die Bezeichnung Sensornetzwerke durchgesetzt, obwohl für die Umsetzung einer Regelung immer mindestens ein Aktuator benötigt wird [64][65]. In der Themenstellung dieser Arbeit steht jedoch die Umsetzung der Regelung über die verteilten Aktuatoren im Vordergrund und es wird daher durchgehend die Begrifflichkeit „Aktuator- und Sensornetzwerk“ verwendet.

## 1.3 Ziel und Gliederung dieser Arbeit

Ausgehend von den Anforderungen der Strömungsregelung durch großräumig verteilte Aktuatoren soll ein echtzeitfähiges Aktuator- und Sensornetzwerk entwickelt werden. Nach abgeschlossener Konzeptentwicklung wird das Netzwerk zunächst als Modell in einer geeigneten Programmumgebung implementiert und anhand von Simulationen evaluiert werden. Schon in diesem Stadium wird das implementierte Modell als Model-in-the-Loop-Simulation im Rahmen der experimentellen Methodenentwicklung in Windkanalversuchen eingesetzt. Dieser Einsatz erfolgt in Zusammenarbeit mit den Projektpartnern des Aerodynamischen Instituts (AIA, RWTH Aachen) und des Instituts für Regelungstechnik (IRT, RWTH Aachen).

Die Generierung der transversalen Wellen auf der Aluminiumoberfläche in den Windkanalexperimenten benötigt hohe Amplituden und Kräfte. Dies kann nicht mit traditionellen Ansätzen wie Piezo-Aktuatoren realisiert werden. Daher wird ein speziell entwickeltes elektromagnetisches Aktuator-System verwendet, welches Oberflächenwellen in bis zu einem Millimeter starken Aluminium im Amplitudenbereich von zehn bis 1000  $\mu\text{m}$  erzeugen kann.

Die Übertragbarkeit der experimentellen Ergebnisse auf die numerischen Simulationen für die weitere Untersuchung in hohen Reynoldszahlbereichen setzt eine hohe Genauigkeit der Aktuierung voraus. Die bei der Variation der

Regelungsvorgaben möglichen lokalen Diskontinuitäten in der Oberflächenaktuierung müssen zum Schutz der Aktuatoren und der Oberfläche ausgeglichen werden. Eine geregelte Erzeugung transversaler Oberflächenwellen in einem inneren Regelkreis schafft somit die Basis für die Methodenentwicklung der aktiven Strömungsbeeinflussung und -regelung in hohen Reynoldszahlbereichen.

Für die Anwendung zur Widerstandsreduktion ist es wichtig, den Energieverbrauch sowohl des Netzwerkes als auch der Aktuatorik beziffern zu können, um eine Abschätzung des Gesamtenergieverbrauchs durchzuführen. Simulationen des Netzwerkmodells mit unterschiedlichen Randbedingungen sollen Unterschiede zwischen den Konfigurationen darlegen und zur Bestimmung einer günstigen Topologie und Aufgabenverteilung führen.

Das weitergehende Ziel dieser Arbeit ist es, durch die Simulation unterschiedlicher Netzwerkkonfigurationen die Spezifikationen für die technische Umsetzung eines mikrocontrollerbasierten, räumlich ausgedehnten Aktuator- und Sensornetzwerkes zu erarbeiten.

Das folgende Kapitel liefert eine Einführung in die Grundlagen der Aktuator- und Sensornetzwerke. Die Kapitel 3, 4 und 5 bilden den Hauptteil dieser Arbeit. In Kapitel 3 werden zunächst die Anforderungen an das Netzwerk zur Verwendung innerhalb eines kaskadierten Regelkreises sowie das Konzept und die Modellimplementierung vorgestellt. Kapitel 4 behandelt das Aktuator-System zur Realisierung der transversalen Oberflächenwellen, die Aktuator-Ansteuerung zur echtzeitigen Änderung der Oberflächenwellen und einen modellbasierten Ansatz zur Wellenregelung. In Kapitel 5 wird die Validierung des Netzwerkes maßgeblich anhand der Modellimplementierung durchgeführt. Es werden zudem Simulationen mit verschiedenen Modellkonfigurationen vergleichend analysiert. Die Strategien zur echtzeitigen Aktuator-Ansteuerung werden validiert und die Integration der Wellenregelung vorgestellt. In Kapitel 6 werden die Ergebnisse diskutiert und ein Ausblick auf weitergehende Forschungs- und Entwicklungsaufgaben gegeben.

Im Rahmen dieses Promotionsvorhabens wurden drei studentische Abschlussarbeiten betreut:

- „Entwicklung einer Real-Time basierten Aktuator-Ansteuerung in LabVIEW“, abgeschlossene Masterarbeit von Sebastian Völkel (Hochschule Niederrhein)
- „Raspberry Pi Based IEEE 1451.1 Network For Distributed Actuation Control“, abgeschlossene Masterarbeit von Lileesha Mulpuri (Hochschule Bremen)

- „Modellbasierte Implementierung klassischer Regelungsstrategien für ein elektromagnetisches Aktuatorsystem in der Turbulenzforschung“, zur Zeit der Fertigstellung dieser Arbeit laufende Bachelorarbeit von Julius Trabert (Hochschule Fulda)



## 2 Grundlagen: Aktuator- und Sensornetzwerke

Die Entwicklung von Kommunikationssystemen hat verschiedene Strukturen und unterschiedliche Vernetzungsstrategien hervorgebracht. Die Teilnehmer in Netzwerken werden auch als *Knoten* bezeichnet. Daten einer Übertragung oder Teile davon werden *Pakete* genannt. Die Vernetzung erfolgt in allen Netzwerken entweder drahtgebunden oder drahtlos. Dabei müssen für die Verwendung drahtloser Netzwerke zusätzliche Einflüsse berücksichtigt werden [4]. Ein großer Vorteil der drahtlosen Vernetzungsstrategie ist die Möglichkeit zur entfernten Platzierung der Knoten im Zusammenspiel mit einem geringen Ressourcenbedarf.

Im Folgenden wird der grundlegende Aufbau von Netzwerken und deren Einsatz in verschiedenen Anwendungsbeispielen anhand der gängigen Standards beschrieben. Insbesondere die Verwendung von Netzwerken in der Regelungstechnik sowie in der industriellen Automatisierung und der Fahrzeugtechnik wird anhand der Literatur geschildert. Außerdem werden Werkzeuge zur Modellierung von Netzwerken und zur analogen Ansteuerung von Aktuator Systemen kurz reflektiert.

### 2.1 Grundlegender Aufbau von Netzwerken

Netzwerke im Allgemeinen wurden entwickelt, um bestimmte Aufgaben zu erfüllen. Rechnernetze im Speziellen sind ein Zusammenschluss selbstständiger elektronischer Systeme wie Computer, Aktuatoren und Sensoren. Diese dienen dem Informationsaustausch und ermöglichen die gemeinsame Nutzung von Ressourcen sowie die zentrale Verwaltung von Daten. Netzwerke ermöglichen die direkte Kommunikation zwischen Teilnehmern beziehungsweise Knoten. Es können zwei oder mehr Teilnehmer daran beteiligt sein. Zum gegenseitigen Verständnis müssen sich diese auf eine „Sprache“ einigen [47]. Innerhalb von Rechnernetzen werden Pakete ausgetauscht, dabei wird das Paket an das

Netzwerk vom Sender übergeben und durch den Empfänger entgegengenommen.

In der Praxis werden zwei Kommunikationsparadigmen in Netzwerken verwendet:

1. Punkt-zu-Punkt-Kommunikation
2. Punkt-zu-Mehrpunkt-Kommunikation

Für die Organisation der physikalischen Verbindung in Netzwerken existieren im Wesentlichen die vier Topologie-Grundformen Bus, Stern, Ring und Baum (siehe Tabelle 2.1). Diese Definitionen stammen aus einer Zeit bevor drahtlose Netzwerke existierten, sie können aber problemlos auf alle möglichen Übertragungsmedien verallgemeinert werden.

**Tabelle 2.1:** Topologie-Grundformen für die Organisation von Knoten in Netzwerken nach [47].

Name	Beschreibung
Bus	Alle Knoten sind mit einem einzelnen Kabelstrang zusammengeschlossen.
Stern	Alle Knoten sind mit jeweils einem Kabel an ein zentrales Gerät, in der Regel ein sogenannter Switch, angeschlossen.
Ring	Alle Knoten sind mit einem einzelnen Kabelstrang verbunden, dieser bildet einen geschlossenen Ring und definiert eine Datenstromrichtung.
Baum	Beschreibt einen Standard für den Zusammenschluss verschiedener Netze, dabei werden Netzwerke verschiedener Topologien an Teilnehmer oder Geräte eines zentralen Netzwerkes angeschlossen.

Für die Datenübertragung zwischen einzelnen Knoten werden sogenannte Protokolle, also einheitliche Abläufe definiert. Die Protokolle werden in einem Protokollstapel organisiert. Diese Idee wird im OSI-Referenzmodell allgemein standardisiert.



lungsschicht stellt sicher, dass die Übertragung in einem einheitlichen Datenformat abgewickelt wird. Gegebenenfalls werden Daten konvertiert, um die korrekte Interpretation der ausgetauschten Informationen zu gewährleisten. Die Kommunikationssteuerungsschicht ist für die Organisation und die Synchronisation der Datenübermittlung zuständig. In der Transportschicht wird die Übertragung der Daten kontrolliert, dies beinhaltet auch Segmentierung und Staukontrolle. Die Vermittlungsschicht übernimmt die Organisation der Paketvermittlung. Auf der Sicherungsschicht wird der Zugriff auf das Übertragungsmedium geregelt und die Übertragungsgeschwindigkeit, der sogenannte Paketfluss, kontrolliert. Die Bitübertragungsschicht regelt die mechanischen, elektrischen und funktionellen Charakteristiken der Übertragung, wie beispielsweise die Bit-Kodierung.

Zwischen den hierarchischen Ebenen innerhalb eines Knotens existiert eine direkte Kommunikation. Eine logische Kommunikation wird auf den einzelnen Schichten zwischen den Knoten realisiert. Das bedeutet, dass die Schnittstellen identisch sein müssen

Das OSI-Referenzmodell bildet die Grundlage für Modelle und Entwicklungen in der Netzwerktechnologie. In [93] sind die Ergebnisse aus dem OSI-Standard zusammengefasst. Mögliche Protokolle werden dort den einzelnen Schichten zugeordnet. Die Auswahl der jeweiligen Protokolle erfolgt anwendungsbezogen. Das OSI-Referenzmodell bildet den Rahmen gängiger implementierter Netzwerkkommunikationen.

### 2.1.2 Das Internet-Referenzmodell

Eine bekannte Anwendung des ISO-OSI-Referenzmodells ist das Internet-Referenzmodell. Die Grundlage dazu bilden die Arbeiten von Braden aus dem Jahr 1989, RFC 1122: [10] und RFC 1123: [11]. Dies ist ein anerkannter Standard zur Internet Kommunikation. Dabei werden jeweils die Schichten sieben, sechs und fünf sowie eins und zwei zusammengefasst. Somit reduziert sich das Schichtenmodell auf vier Kommunikationsschichten. Für die Internetkommunikation sind verschiedene Kommunikationsprotokolle festgelegt. Einige Beispiele sind in Abb. 2.2 dargestellt.

Applikationen wie Webbrowser verwenden diesen Protokollstapel. Das *Hypertext Transfer Protocol* (HTTP) dient der Übertragung von Webseiten. Das *File Transfer Protocol* (FTP) ist für die Übermittlung von Dateien und das *Simple Mail Transfer Protocol* (SMTP) ist für die Übertragung von E-Mails definiert.

7	Anwendung	HTTP, FTP, SMTP
6	Darstellung	
5	Kommunikations- steuerung	
4	Transport	TCP, UDP
3	Vermittlung	ARP, IP
2	Sicherung	IEEE 802, Ethernet, Token-Ring
1	Bitübertragung	

**Abb. 2.2:** Das Internet-Referenzmodell implementiert das OSI-Referenzmodell. Den teilweise kombinierten Ebenen lassen sich die verbreiteten Protokolle zuordnen.

Auf der Transportschicht werden sowohl verbindungsorientierte als auch verbindungslose Protokolle eingesetzt. Das verbindungsorientierten Protokoll mit der größten Verbreitung ist das *Transmission Control Protocol* (TCP) [68]. Die ersten Ideen zur Entwicklung wurden von Cerf und Kahn in den frühen 1970er Jahren publiziert. Der TCP-Standard im RFC 793 wurde 1981 veröffentlicht. In dieser Klasse der Protokolle wird vor Beginn der Kommunikation eine Verbindung durch das Netzwerk zum Kommunikationspartner etabliert. Die Daten werden in der Reihenfolge zugestellt, in der sie abgesendet werden. Außerdem wird der Erhalt der Pakete bestätigt. Bei der Verwendung verbindungsloser Protokolle wie beispielsweise des gängigen *User Datagram Protocols* (UDP) [67] werden die Daten möglicherweise in anderer Reihenfolge als in der gesendeten zugestellt und auch nicht bestätigt. Dadurch besteht die Möglichkeit, dass Pakete verloren gehen. Das Protokoll selbst wurde als Alternative zu TCP mit minimalem Protokoll-Overhead entwickelt und 1980 im RFC 768 veröffentlicht. Innerhalb der Transportschicht oder darunter finden sich, je nach Protokoll, Mechanismen wie Segmentierung, Fluss- und Staukontrolle wieder. Diese sind für die Aufteilung der Nachricht in Pakete sowie die dazu notwendige Vermeidung von einem Paketstau und einem daraus resultierenden Verlust beim Empfänger zuständig.

Innerhalb der Vermittlungsschicht werden Protokolle wie das *Address Resolution Protocol* (ARP) oder das *Internet Protocol* (IP) zur Adressierung und zur Weitergabe von Paketen zwischen der Transportschicht und der untersten Schicht verwendet. Das IP-Protokoll wurde parallel zum TCP als verbindungsloses Protokoll zur Verteilung von Datenpaketen zwischen Quell- und Zielrechner entwickelt und 1981 standardisiert [44]. Die Teilnehmer der Kommunikation werden durch eine definierte Adresse, der sogenannten IP-Adresse, identifiziert. Es bildet die Grundlage für die heutige Internetkommunikation.

Der IEEE 802 Standard definiert die Aufteilung der Sicherungsschicht in die Subschichten *Medium Access Control* (MAC) und *Logical Link Control* (LLC) [37]. MAC beinhaltet den Zugang zu lokalen Netzen, während LLC für Adressierung und Fehlerprüfung zuständig ist. Einige Protokolle sind in IEEE 802 definiert, dazu gehören Token-Ring und das weit verbreitete Ethernet. Diese spezifizieren Kabeltypen, Signaldarstellung, Paketformate und Kommunikationsprotokoll für die MAC-Schicht.

### 2.1.3 Echtzeitfähige Protokolle

Eine Kommunikationsmethode, die Daten innerhalb eines fest zugesagten Zeitraums überträgt, wird als echtzeitig bezeichnet. Nach [85] wird dies durch die Verminderung der Latenzzeiten erreicht, beispielsweise durch eine deterministische Paketverteilung und Fehlerbehandlung. Die Kollisionsvermeidung auf dem Übertragungsmedium ermöglicht die Zusicherung einer maximalen Übertragungsdauer des versendeten Pakets. Der Determinismus wird beispielsweise durch die Weitergabe eines Tokens oder durch eine Master-Slave-Strategie sichergestellt, bei der ein Teilnehmer im Netzwerk die Kommunikation der anderen Teilnehmer kontrolliert.

Es gibt verschiedene echtzeitige Netzwerkprotokolle für unterschiedliche Anwendungen und mit unterschiedlichen Antwortzeiten. Im Folgenden sind einige Beispiele aufgezeigt. Die Protokolle beschreiben häufig nicht nur eine Ebene des OSI-Referenzmodells, sondern verbinden die unteren Schichten direkt mit der Applikationsschicht [32], gegebenenfalls mit einer zusätzlichen Middleware. Einige Protokolle unterliegen technischen Beschränkungen und können nur mit spezieller und zumeist sehr teurer Hardware betrieben werden.

Das *Realtime Transport Protocol* (RTP) arbeitet auf der Anwendungsschicht des Internet-Referenzmodells und verwendet das UDP-Protokoll auf der

Transportschicht. Das RTP-Protokoll wurde 1996 standardisiert und ist seit 2003 im RFC 3550 Standard verfügbar [77]. Es dient zur echtzeitigen Übertragung von Datenströmen beispielsweise durch das Internet. Ein weiteres Protokoll der Anwendungsschicht ist *Modbus/TCP*. Als Protokoll zur Kommunikation mit speicherprogrammierbaren Steuerungen entwickelt, wurde es 2007 in die Norm IEC 61784 aufgenommen [40]. Es verwendet das TCP/IP-Protokoll zur Datenübertragung.

Die echtzeitige Kommunikation auf den untersten Schichten des OSI-Referenzmodells wird häufig durch Feldbus-Systeme beschrieben. Im IEC 61784 Standard [40] sind einige dieser Systeme zusammengefasst, darunter auch *PROFIBUS*. Dieses Protokoll wurde als Gemeinschaftsprojekt von 21 Firmen und Instituten, darunter AEG, Bosch, Siemens sowie das WZL der RWTH Aachen, zur Prozessautomatisierung entwickelt. Die vom BMBF geförderte Entwicklung begann Ende der 1980er Jahre. Heute sind verschiedene Varianten des Protokolls verfügbar, auf die aber hier nicht näher eingegangen wird. Das middlewarebasierte *PROFINET* und das Master-Slave-basierte *Powerlink* sind ebenfalls im IEC 61784 Standard referenziert. Ein weiteres Protokoll für den Echtzeitbetrieb ist das *Controller Area Network (CAN)*. Es wurde 1989 als Bussystem für die Vernetzung von Steuergeräten in der Fahrzeugtechnik von Bosch entwickelt und ist in der ISO 11898 Norm [41] standardisiert. Die Entwicklung des FlexRay-Protokolls wurde im Jahr 2000 von den Unternehmen BMW, Daimler AG, Philipps und Motorola gestartet. Das Protokoll wurde 2010 in den ISO 17458 Standard [42] überführt.

Die echtzeitfähigen Kommunikationssysteme auf den untersten Kommunikationsschichten verwenden speziell zugeschnittene Anwendungsprotokolle, die direkt auf das echtzeitfähige Protokoll zugreifen und so eine Datenübertragung innerhalb einer vorgegebenen Zeit ermöglichen.

### 2.1.4 Protokolle der Anwendungsschicht

Im Zusammenhang mit dem Internet-Referenzmodell wurden bereits Protokolle der Anwendungsschicht wie HTTP oder FTP vorgestellt. Für die echtzeitige Übertragung wurden die Protokolle RTP und Modbus/TCP beschrieben.

Für die Vernetzung von Aktuatoren und Sensoren ist eine Protokollfamilie in den *Smart Transducer Interface Standards* im IEEE 1451 [38] definiert.

Die Entwicklung dieser Protokollfamilie begann 1993 und wurde vom technischen Komitee für Sensortechnologie der *IEEE Instrumentation and Measurement Society* durchgeführt. Das Ziel war die Definition verschiedener allgemeiner Kommunikationsschnittstellen um die Vernetzung von Aktuatoren, Sensoren, Instrumenten und beispielsweise industriellen Netzwerken zu ermöglichen. Der Zugriff auf die einzelnen Komponenten sollte durch lokale Netzwerke und über das Internet ermöglicht werden. Der Standard bietet ein Framework für die einheitliche Kommunikation zwischen Aktuatoren, Sensoren und Reglern und ermöglicht den sofortigen Einsatz und die Bereitstellung aller Funktionalitäten nach der Verbindung eines Knotens mit dem Netzwerk. Diese Funktionalitäten sind ein wichtiges Element im *Internet of Things* (IoT) beziehungsweise in *Cyber-Physischen-Systemen* (CPS) [38].

**Tabelle 2.2:** Definitionen innerhalb des IEEE 1451 Smart Transducer Interface Standards.

Name	Beschreibung
IEEE 1451.0	Einheitliche Operationen und das Format der TEDS
IEEE 1451.1	Definition von Objektmodell, Datenmodell und Kommunikationsmodell
IEEE 1451.2	Schnittstellen zwischen NCAP und TIM für Punkt-zu-Punkt Verbindungen
IEEE 1451.3	Schnittstellen zwischen NCAP und TIM für einen TIM-Verbund
IEEE 1451.4	Gemischte analoge Schnittstelle für TIM die analog und digital arbeiten
IEEE 1451.5	Schnittstelle zwischen NCAP <sub>C</sub> und TIM für drahtlose Kommunikation
IEEE 1451.6	Schnittstelle für die Verwendung von CANopen zwischen NCAP und TIM
IEEE 1451.7	Schnittstelle zwischen TIM und RFID-Systemen

Der IEEE 1451 Standard basiert auf dem sogenannten *Network Capable Application Processor* (NCAP) Modul und dem *Transducer Interface Module* (TIM). Die Strategie des IEEE 1451 ist die Vernetzung des TIM, der einen Sensor oder Aktuator definiert, über den NCAP mit der Außenwelt. Der NCAP liefert die nötige Intelligenz und das TIM die notwendige Mess- oder

Steuertechnik. Der Zusammenschluss eines NCAP mit einem TIM wird als *Smart Transducer* beschrieben. Ein weiterer zentraler Teil des IEEE 1451 ist das *Transducer Electronic Data Sheet* (TEDS). Es beinhaltet Identifikationsdaten des TIM, die dieses dem NCAP in einer standardisierten Form zur Verfügung stellt.

Innerhalb der Protokollfamilie werden verschiedene Substandards definiert. Diese sind in Tabelle 2.2 aufgeführt. Einen detaillierten Überblick über die einzelnen Aufgaben und Funktionsweisen bieten Song und Lee in [79]. Ein Beispiel für eine IEEE 1451 kompatible Kommunikation ist in [34] beschrieben.

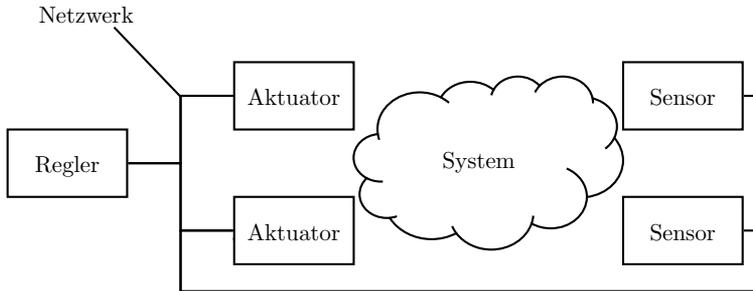
## 2.2 Aktuator- und Sensornetzwerke

Die Aufteilung von Datenverarbeitung, Regelung und Kommunikation in verschiedene Ebenen des Maschinenbetriebs wird heute praktiziert [91]. Dabei etabliert sich die Verwendung räumlich verteilter Aktuatoren und Sensoren. Weiterhin stößt die bisher eingesetzte Punkt-zu-Punkt Kommunikation zwischen Reglern, Aktuatoren und Sensoren, beispielsweise in der industriellen Automatisierung an ihre Grenzen. Durch den Verfall der Preise bei der entsprechenden Hardware werden immer mehr Regelkreise mithilfe der Kommunikation zwischen Computereinheiten geschlossen. Dabei handelt es sich um sogenannte vernetzte Regelsysteme. Diese sind sowohl in der industriellen Automatisierung, als auch in der Fahrzeugtechnik und in analogen Anwendungen zu finden. Damit wird die Effizienz verbessert und sowohl Kosten als auch Zeit für Wartung reduziert. Jedoch bringt diese Umstellung nicht nur Vorteile, sondern impliziert auch Nachteile wie beispielsweise Unsicherheiten oder zusätzliche variable zeitliche Verzögerungen [91].

### 2.2.1 Netzwerke in der Regelungstechnik

Abb. 2.3 zeigt eine räumliche Trennung zwischen Regler, Aktuatoren und Sensoren. Der Datenaustausch zur Steuerung der Aktuatoren und zum Auslesen der Sensoren erfolgt über ein gemeinsames Netzwerk.

Neben dem Einsatz vernetzter Regelsysteme in der industriellen Automatisierung und in der Fahrzeugtechnik werden reine Sensornetzwerke für die Überwachung von Gebäuden, technischen Anlagen und Verkehrssystemen auf dem



**Abb. 2.3:** Schematischer Aufbau eines Aktuator- und Sensornetzwerkes zur Verwendung in einem vernetzten Regelsystem.

Forschungs- und Anwendungsgebiet „Structural Health Monitoring“ verwendet [8].

Brindha et al. bezeichnen in [13] die Untersuchung und Verwendung von Aktuator- und Sensornetzwerken in vernetzten Regelsystemen als ein interdisziplinäres Feld. Für den Einsatz von Netzwerken zur Regelung eines Systems mithilfe verteilter Aktuatoren müssen diese hinsichtlich der Eigenschaften wie beispielsweise der induzierten Verzögerungszeit untersucht und optimiert werden. Ein Aspekt ist die Entwicklung und der Einsatz spezieller Protokolle um die beschränkte Rechenleistung optimal auszunutzen [59].

Es gibt verschiedene Netzwerkklassifizierungen bis hin zum weltweiten Netzwerk (*Global Area Network* (GAN)) [47]. Aktuator- und Sensornetzwerke für die Verwendung in vernetzten Regelsystemen lassen sich in die Klasse der lokalen Netzwerke (*Local Area Network* (LAN)) einordnen. Im Folgenden werden ausschließlich diese betrachtet.

In [91] wird der Einfluss von Netzwerken und die Möglichkeiten für eine verteilte Regelung auf Basis der zumeist verwendeten zentralen oder dezentralen Regelung beschrieben. Die zentrale Regelung verwendet dabei einen einzelnen Regler, der direkt mit allen Sensoren und Aktuatoren verbunden ist. Die dezentrale Regelung bildet Subsysteme, die jeweils über einen eigenen Regelkreis verfügen, aber keine Verbindungen zwischen den einzelnen Komponenten zulassen. Nicht jedes System kann durch eine dezentrale Regelung stabilisiert werden. Die Verbesserung der Regler-Performanz kann laut [91] durch eine quasi-dezentrale Regelung erreicht werden, bei der die Regelung und Datenverarbeitung sowohl dezentral, als auch in einem zentralen externen Regler stattfindet. Als Beispiel wird dazu die Regelung elektrischer

Stromversorgungssysteme angeführt. Hierbei handelt es sich um großflächige, verteilte Systeme, die viele Regelungsaufgaben beinhalten. Diese werden bisher mithilfe dezentraler Regler betrieben. Um eine robustere Regelung und großflächige Synchronisation dieser Systeme zu erreichen, wird vorgeschlagen, quasi-dezentrale Regelungen zu implementieren [91].

### **2.2.2 Netzwerke in der industriellen Automatisierung und der Fahrzeugtechnik**

Zwei bekannte Themenfelder für den Einsatz von Aktuator- und Sensornetzwerken sind die industrielle Automatisierung und die Fahrzeugtechnik. Je nach Einsatzgebiet stellen die Applikationen unterschiedliche Anforderungen an die Echtzeitfähigkeit und auch an weitere Eigenschaften wie Fehlertoleranz, Bandbreite und Flexibilität [65].

In der Fahrzeugtechnik sind die Protokolle CAN als bisher meist verwendeter Standard und FlexRay als zukünftiger Standard in der Echtzeitkommunikation innerhalb von Fahrzeugen zu benennen [65]. Beide sind auf den zwei untersten Ebenen des OSI-Referenzmodells angesiedelt und benötigen speziell zugeschnittene Anwendungsprotokolle. CAN ist ein leistungsstarkes, eventgesteuertes Protokoll. Es existiert in verschiedenen Konfigurationen. Je nach Anwendung wird Wert auf eine höhere Fehlertoleranz oder eine hohe Geschwindigkeit gelegt. Die Anforderungen sind im Zusammenhang mit der „x-by-wire“-Technologie gestiegen, darunter versteht man die elektronische Steuerung bisheriger hydraulischer Systeme wie Bremsen oder Lenkung. Diese hochsensitiven zeitlichen Anforderungen sowie der bisherige ereignisbasierte Datenverkehr werden im FlexRay-Protokoll vereint. Für die Applikation in der Fahrzeugtechnik werden zusätzlich spezielle Protokolle auf der Anwendungsschicht eingesetzt.

Nach der Einführung der Produktionsmaschinen in der ersten und dem Beginn der Massenproduktion mit elektrischer Energie in der zweiten industriellen Revolution werden die Feldbussysteme seit der dritten industriellen Revolution in der Automatisierung eingesetzt. Die voranschreitende Integration des Ethernet-Standards ist ein Indiz für die herannahende vierte industrielle Revolution, die auch unter dem Namen „Industrie 4.0“ bekannt ist und eng mit dem Begriff der Cyber-Physischen-Systeme verbunden ist. Innerhalb dieses Prozesses wird die Vernetzung der Produktionssysteme verschiedener Produktionsstandorte vorangetrieben, sodass der Zugriff von überall ermöglicht wird und Fertigungsmaschinen Informationen über den eigenen Zustand

im Netzwerk vorhalten. Wichtige Begriffe in dem Zusammenhang sind sowohl die Stabilität als auch die Privatsphäre der Daten und deren sichere Übermittlung [24].

Einige Beispiele für verwendete Protokolle innerhalb der industriellen Automatisierung sind die Systeme Modbus, PROFINET und Powerlink [64]. Während Modbus auf der Applikationsschicht angesiedelt ist und über das TCP/IP-Protokoll kommuniziert, arbeiten PROFINET und Powerlink mit einer Middleware oberhalb der MAC-Ebene. In diesen Bereichen werden Ethernet-basierte Netzwerke eingesetzt. Modbus als Anwendungsprotokoll wird in der Prozessautomatisierung mit geringen Echtzeitanforderungen verwendet. PROFINET und Powerlink bieten Zykluszeiten von einer Millisekunde bis zu  $250 \mu\text{s}$  für den Einsatz in Regelungssystemen für Bewegungen, zum Beispiel in der Robotik. PROFINET kann flexibel auf die Anwendung abgestimmt werden. Es ermöglicht viele verschiedene Konfigurationen, um den Anforderungen der industriellen Verwendung zu genügen. Powerlink ist ein offener Standard und erlaubt die gleichzeitige Übertragung von Echtzeitdaten und Datenpaketen gängiger Transportprotokolle. Diese Konfiguration schafft eine hohe Flexibilität für den Einsatz in Automatisierungssystemen.

### 2.3 Netzwerkknoten

Die Entwicklung von kleinen und günstigen, kommerziellen, mikrocontrollerbasierten Plattformen für die Verwendung mit verschiedenen etablierten Betriebssystemvarianten schreitet zügig voran. Ein Beispiel dafür ist der Raspberry Pi [70]. Er ist ein kreditkartengroßer Computer mit Beschränkungen in Energieverbrauch und Rechenkapazität. Er ist seit Anfang 2012 kommerziell erhältlich. Das System arbeitet mit einem Linux-basierten Betriebssystem. Der Raspberry Pi kann einfach und schnell konfiguriert werden und unterliegt einer ständigen Weiterentwicklung. Er kann auch für den Einsatz in der Datenerfassung verwendet werden [9]. Als Mikrocontroller-Plattform ist der Raspberry Pi eine günstige Kompromisslösung, bietet jedoch kein Echtzeitbetriebssystem und generiert durch das Linux-System einen nicht unbeachtlichen Overhead. Aufgrund des Preis-Leistungs-Verhältnisses ist diese Plattform eine gute Option für Experimente in Lehre und Forschung, die keine speziell zugeschnittene Hardware benötigen.

Als Teilnehmer innerhalb des Aktuator- und Sensornetzwerkes für den Einsatz in der Strömungsregelung ist der *intelligent Network Operating Device*, kurz iNODE [75][74], vorgesehen. Er ist eine Eigenentwicklung des ZEA-2.

Dieser wurde bereits in verschiedenen Forschungsprojekten mit den folgenden Schwerpunkten eingesetzt:

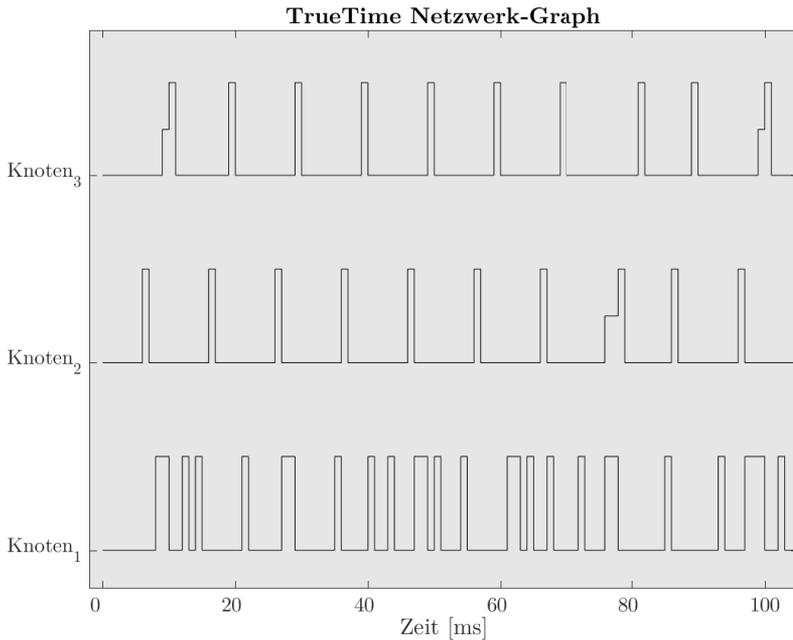
- Bioelektronik und Sensorik [75]
- Neurophysiologie [75]
- Nicht-Invasive Rehabilitation [92]
- Vagus-Nerv-Stimulation zur Blutdruckregelung [21]
- Retina-Implantat-Entwicklung [74]

Diese generische Mikrocontroller-Plattform verfügt in seiner Grundkonfiguration über zwei analoge Ausgänge und bis zu acht analoge Eingänge. Die Plattform kann flexibel auf die spezifischen Anforderungen unterschiedlicher Anwendungen angepasst werden. Bei der Auswahl der Komponenten kann Rechenleistung, Speicherkapazität und die Energieversorgung flexibel aufeinander abgestimmt werden. Der vorgesehenen Anwendung entsprechend besteht die Möglichkeit, auf die analogen Schnittstellen über eine Netzwerkanbindung zuzugreifen. Die Auswahl der Komponenten wird durch den gewünschten Formfaktor begrenzt. Der iNODE ist auf niedrigen Energieverbrauch optimiert. Mithilfe des angepassten Betriebssystems lässt sich eine echtzeitige Datenverarbeitung garantieren. Dem stehen allerdings die hohen Entwicklungskosten und die aufwändige Implementierung gegenüber.

## 2.4 Werkzeuge zur Modellierung und Ansteuerung

Im Folgenden werden Werkzeuge zur Modellierung von Netzwerken und für die Ansteuerung von Aktuator Systemen und Auslese von Sensoren vorgestellt.

Die Modellierung und der Einfluss von Netzwerken in vernetzten Regelsystemen ist sowohl analytisch als auch ereignisbasiert möglich. Am Beispiel von FlexRay wird in [88] das Netzwerk formalisiert und die Einflüsse des Netzwerkes auf einen ohne Netzwerk ausgelegten Regler untersucht. Ein spezielles Augenmerk liegt dabei auf der Zeitplanung der Datenübertragung des Netzwerkprotokolls, die in verschiedene Kommunikationsabschnitte unterteilt ist. Mithilfe dieses Ansatzes wird die minimal notwendige Paketrate für die Stabilisierung des Regelkreises bestimmt.



**Abb. 2.4:** Die Ausgaben der TrueTime Netzwerk-Blöcke zeigen die Netzwerkauslastung, diese wird für jeden Knoten differenziert dargestellt. Die dreigeteilte Höhe jedes Graphen zeigt die aktuelle Kommunikationsphase an und unterscheidet zwischen „Unten“ (keine Kommunikation), „Mitte“ (Warten auf Übertragung) und „Oben“ (Medium ist in Verwendung).

Die ereignisbasierte Emulation von Kommunikationsalgorithmen wird zum Beispiel in MATLAB/Simulink durch die TrueTime Toolbox ermöglicht [16]. Diese Toolbox gestattet die Vernetzung von Regler und System mit unterschiedlichen Netzwerkprotokollen. Dazu werden die Kommunikation sowie komplexe Anwendungen auf simulierten echtzeitfähigen, multiprogrammbetriebenen, eingebetteten Systemen implementiert. Dies erfolgt durch die Verwendung von Netzwerk- und sogenannten Kernel-Blöcken. Für implementierte Anwendungen können sowohl Prioritäten als auch Zeitsteuerung konfiguriert und Rechen- und Wartezeiten vorgegeben werden. Die Verwendung von bereits ausgelegten Reglern in Simulink innerhalb der Applikationen ist weiterhin möglich. Der Einfluss von Asynchronitäten zwischen verschiedenen Anwendungen und Netzwerknoten kann im Modell integriert und in der Simulation evaluiert werden. Netzwerke werden hinsichtlich MAC-Protokoll, Übertragungsrates und weiteren Parametern konfiguriert [17]. Durchgeführte Simulationen lassen sich anhand der Ausgabegraphen im Hinblick auf Zeitverteilung der einzelnen Anwendungen und Netzwerkereignisse analysieren.

Abb. 2.4 zeigt den Ausschnitt des Ausgabegraphen eines TrueTime Netzwerk-Blockes aus dem der Toolbox mitgelieferten Beispiel „networked.mdl“. Diese Graphen werden auch von den Kernel-Blöcken ausgegeben und zeigen für jeden Knoten im Netzwerk oder einen Prozess im Kernel eine Zeitverteilung an. Im Beispiel übertragen drei verschiedene Knoten auf dem gleichen Netzwerk Daten. Die Ausgaben im Graphen sind jeweils dreigeteilt in niedrig, mittel und hoch und zeigen an, ob sich der Sendevorgang oder der Prozess im Leerlauf oder im Wartezustand befinden oder ob das Medium oder der Prozess blockiert sind [17].

Einige Beispiele zur Verwendung der TrueTime-Toolbox sind in [18] beschrieben. Es wird ein kleines Netzwerkmodell mit vier Knoten, einem Aktuator, einem Sensor und einem Regler sowie einem Knoten, der eine Störung im Netzwerk erzeugt, untersucht und die Simulationsergebnisse werden erläutert. Die Möglichkeit des Einsatzes sowohl zeit-, als auch ereignisgesteuerter Regler wird verdeutlicht. Als Anwendungsbeispiel wird die Kommunikation zwischen mobilen, sich bewegenden Knoten und die Möglichkeit zum Testen von Algorithmen zur Routenplanung der Pakete vorgestellt. Als Resultat wird TrueTime als flexible Toolbox zur Untersuchung von Netzwerken hinsichtlich unterschiedlicher Gesichtspunkte präsentiert. Dabei existiert die Möglichkeit, eine vorgegebene Anwendung auf einem eingebetteten System, die Zeitsteuerung innerhalb der Knoten, drahtgebundene und drahtlose Kommunikation sowie die Aktuator-, Sensor- und Systemdynamik innerhalb eines Modells zu analysieren. Zudem lassen sich Umgebungseinflüsse und auch der Energieverbrauch einzelner Komponenten modellieren und simulieren.

Die Toolbox wird vorwiegend verwendet, um die Einflüsse von Netzwerken auf die Separierung von Regler und System in Regelkreisen zu untersuchen [12]. Dabei werden vorwiegend Feldbussysteme betrachtet. In [12] wird ein Vergleich zwischen CAN und einem switched Ethernet durchgeführt. Für den Vergleich werden Netzwerke unter optimalen Bedingungen sowie mit Informationsverlusten und Störungen durch Verwendung der Netzwerkinfrastruktur innerhalb einer zusätzlichen Applikation herangezogen. Das Verhalten des Regelkreises hängt schlussendlich vom verwendeten MAC-Protokoll und dessen Konfiguration sowie dem Umfang der externen Einflüsse ab.

Der Einsatz einer Netzwerksimulation innerhalb einer Model-in-the-Loop-Umgebung erfordert analoge Aus- und Eingänge zur Ansteuerung von Aktuatoren und zum Auslesen von Sensoren. Für die Verbindung mit analogen Aus- und Eingängen stehen verschiedene Werkzeuge zur Verfügung.

MATLAB und Simulink stellen hierfür Toolboxes bereit, die sowohl in der Entwicklungsumgebung als auch auf einem Echtzeitsystem verwendbar sind. Gängige Hardware wird unterstützt, jedoch sind die Funktionalitäten einiger Schnittstellen, wie beispielsweise Pufferspeicher, nicht ausnutzbar. Eine versionsübergreifende Kompatibilität für Hardware-Schnittstellen ist nur sehr eingeschränkt gegeben.

Nach [82] sind MATLAB und Simulink flexibel für Design und Analyse technischer Applikationen einsetzbar. Es werden auch Studien zur Verwendung von MATLAB und LabVIEW im Verbund beschrieben. Dabei wird ein Modell des Systems und die Auslegung der Regelung mithilfe von MATLAB und die Implementierung der Software auf einem Echtzeitsystem mithilfe von LabVIEW realisiert. Für hohe Sampleraten und die Automatisierung ist LabVIEW besser als MATLAB geeignet.

He beschreibt in [36] die gute Hardwareanbindung von LabVIEW und die Möglichkeiten zur Kommunikation. Dabei besitzt MATLAB viele Toolboxes und kann komplexe Anwendungen besser bewältigen. Außerdem ist die Modellierung und Offline-Datenverarbeitung effizienter.

In dieser Arbeit wurden MATLAB/Simulink und TrueTime für die Modellierung und Simulation des Netzwerkes und der darin integrierten Anwendungen verwendet. Die Ansteuerung der Aktuatorik und Sensordatenerfassung wurde mit LabVIEW umgesetzt.

# 3 Netzwerkkonzept des kaskadierten Regelkreises

Das Netzwerkkonzept zur Anwendung in der Strömungsregelung wurde in Abstimmung mit den Projektpartnern der Forschergruppe FOR1779 entwickelt. Für die Anknüpfungspunkte des Aerodynamischen Instituts (AIA) und des Instituts für Regelungstechnik (IRT) der RWTH Aachen wurden einerseits die spezifischen Projektanforderungen und andererseits die absehbaren Anforderungen der Regelung einer turbulenten Strömung mittels großflächig verteilter Aktuatoren berücksichtigt. Das Netzwerk ist entsprechend als kaskadierter Regelkreis [2] realisiert, bestehend aus einem äußeren Regelkreis mit einem externen Regler, der Strömungsregelung, einer Steuerungskomponente für die Aktuierung der umströmten Oberfläche (Wellengenerierung und -änderung) und einem inneren Regelkreis zur Regelung der Oberflächenwellen (Wellenregelung). Das Konzept ist dabei flexibel ausgelegt gegenüber Änderung der Applikationsanforderungen. Die Validierung der Übertragung und Umsetzung der Stellgrößen der externen Regelung erfolgt dabei durch ein Netzwerkmodell und kann Model-in-the-Loop an die Aktuator-Ansteuerung, die auch die Wellenregelung beinhaltet, angekoppelt werden.

Im Folgenden werden die Spezifikationen für das Netzwerk basierend auf den allgemeinen und den applikationsspezifischen Anforderungen beschrieben und das entwickelte Netzwerkkonzept sowie das in MATLAB/Simulink implementierte Modell dieses Aktuator- und Sensornetzwerkes vorgestellt. Die Inhalte dieses Kapitels finden sich zu unterschiedlichen Teilen in diesen Veröffentlichungen von Dück et al.: [26], [28], [27].

## 3.1 Spezifikation des Netzwerkes

Die zentrale Aufgabe des Netzwerkes ist die großflächige Umsetzung der Regelungsvorhaben der externen Regelung mit einer entsprechend der Applikationsanforderungen zu definierenden Genauigkeit und zeitlicher Präzision.

Eine Randbedingung hierbei ist die energieeffiziente Realisierung der Vorgaben, sodass der Energieverbrauch für Berechnungen, Datenübertragung und Aktuierung geringer ist als die erzielte Energieersparnis durch die Oberflächenaktuierung. Im Gegensatz zu einem einzelnen zentralen Rechensystem bietet eine Verteilung der Rechenleistung Vorteile, insofern dass in den verteilten Recheneinheiten Daten parallel verarbeitet werden können, die Mess- und Aktuierungssignale kürzere Übertragungswege benötigen und die Recheneinheiten sich im Prinzip auch bei Bedarf gegenseitig ersetzen können.

#### **3.1.1 Allgemeine Anforderungen für großflächige Netzwerke**

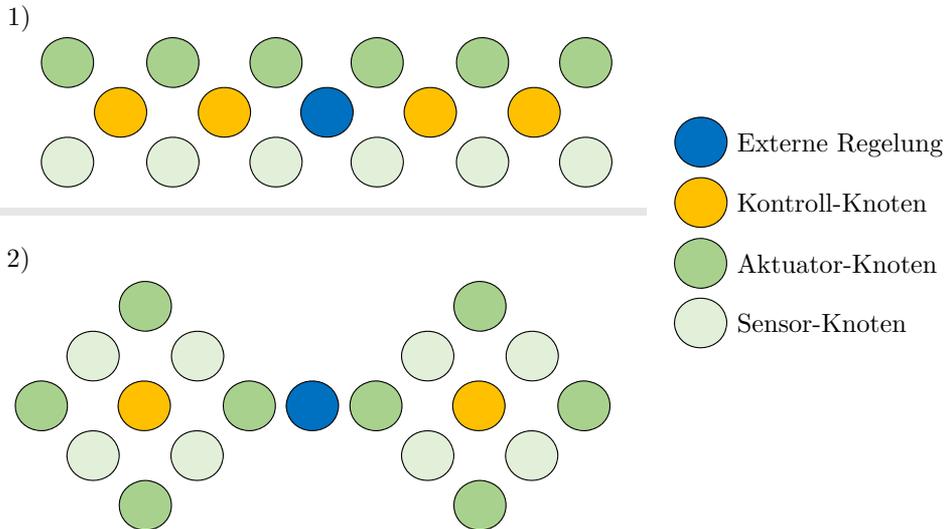
In großen Netzwerken mit vielen verteilten Sensoren und Aktuatoren ist die induzierte Latenzzeit zwischen Anfrage und Antwort ein beschränkender Faktor für die Anwendung. In echtzeitfähigen Netzwerken müssen benötigte Daten in einem gewissen Zyklus verarbeitet werden. Deterministische Protokollabläufe stellen eine maximale Übertragungs- beziehungsweise Antwortzeit sicher. Dabei sollten sehr große Datenpuffer, wie sie zum Beispiel beim echtzeitigen *On Demand Video Streaming* eingesetzt werden, aus Effizienzgründen vermieden werden.

Allgemein lässt sich das Netzwerk für ein vernetztes Regelsystem in einem kaskadierten Regelkreis als Kombination einer zentralen und einer dezentralen Regelung gemäß [91] beschreiben. Die Teilnehmer im Netzwerk lassen sich in drei Klassen aufteilen:

1. Kontroll-Knoten
2. Aktuator-Knoten
3. Sensor-Knoten

Diese Einteilung bietet einen generischen Ansatz für Netzwerke zur Realisierung kaskadierter Regelkreise. Die Kontroll-Knoten übernehmen die Verteilung der Parameter für die Aktuatorik und Verarbeitung und Synthese der digitalisierten Sensordaten. Die Aufgabe der Aktuator-Knoten ist die Umsetzung der Vorgaben der zentralen Regelung, wobei diese Umsetzung über die dezentrale Regelung (innerer Regelkreis) basierend auf lokal erfassten Sensordaten erfolgt. Die Sensor-Knoten übernehmen die Erfassung und Vorverarbeitung von Messdaten und deren Bereitstellung an die externe Regelung. Da die Sensordaten bereits innerhalb des Netzwerkes durch die Sensor-Knoten

und Kontroll-Knoten verarbeitet werden, kann die externe Regelung auch als quasi-dezentrale Regelung nach [91] bezeichnet werden. Es muss nicht nur



**Abb. 3.1:** Zwei verschiedene Aufteilungen von großflächigen Aktuator- und Sensornetzwerken zeigen die räumliche Anordnung und die möglichen unterschiedlichen Verhältnisse von Sensor-Knoten, Aktuator-Knoten und Kontroll-Knoten zu einer externen Regelung.

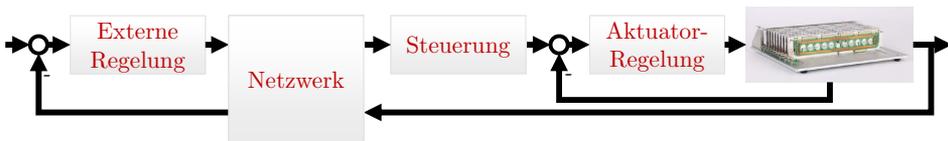
Die Größe des Netzwerkes bezieht sich auf die gesamte Anzahl der Teilnehmer beziehungsweise auf die Anzahl der Aktuatoren und Sensoren innerhalb der Anwendung. Zunächst werden als Mindestanforderung 100 unabhängig ansteuerbare Aktuatoren festgelegt. Mit der Auslegung der einzelnen dezentralen Recheneinheiten für zwei analogen Aktuatoren ergibt sich damit eine Mindestzahl von 50 Knoten zur Ansteuerung. Das verwendete Konzept soll eine einfache Skalierung auf größere Knotenanzahlen unterstützen. Für die Realisierung der gewünschten Regelungsvorhaben durch eine aktuatornahe Steuerung und Regelung wird eine angemessene Verteilung der Rechenleistung in Kombination mit einer günstigen Topologie und einer robusten Kommunikation angestrebt. So kann die echtzeitige Umsetzung von Regelungsvorhaben garantiert werden [85].

Das Netzwerkkonzept muss Optimierungsmöglichkeiten hinsichtlich Größe

und Verteilung der einzelnen Knoten bieten. Die Möglichkeit unterschiedlicher räumlicher Verteilungen (siehe Abb. 3.1) muss dabei berücksichtigt werden. Die Optimierung von Reaktionszeiten und Strategien zur Sicherstellung der Ausfallsicherheit sowie zur Reduktion der Gesamtkosten und Optimierung des Energieverbrauchs müssen ebenfalls ermöglicht werden.

#### 3.1.2 Applikationsanforderungen

Die Umsetzung der Strömungsbeeinflussung und -regelung mithilfe transversaler Oberflächenwellen auf einer ausgedehnten Fläche lässt sich als kaskadierter Regelkreis beschreiben (siehe Abb. 3.2) und ist eine sehr anspruchsvolle Beispielanwendung für den Einsatz eines Aktuator- und Sensornetzwerkes. Das Konzept und die konkrete Algorithmik für die externe Strömungsregelung wird vom Projektpartner IRT entwickelt. Da somit die genauen Vorgaben für das Netzwerk noch nicht spezifiziert werden können, werden das Netzwerkkonzept und die Modellimplementierung möglichst flexibel ausgelegt. Diese flexible Auslegung erlaubt auch eine einfache Übertragung des Netzwerkes



**Abb. 3.2:** Die schematische Darstellung verbindet die externe Regelung mit dem Netzwerkmodell, einer Steuerung und einer Aktuator-Regelung in einem kaskadierten Regelkreis. Das Netzwerk übernimmt die Aufgabe der Datenübertragung. Die Steuerung sorgt für eine reibungslose Überführung der Stellgrößen der externen Regelung auf das Aktuator-System und die Aktuator-Regelung stellt die dauerhafte, korrekte Realisierung der vorgegebenen Stellgrößen beziehungsweise einer daraus resultierenden Bewegung sicher.

In einem Grundkonzept zum Anwendungsaufbau werden die benötigten Komponenten zunächst zu einem Framework zusammengefasst [26]. Dieses Framework beinhaltet die Schnittstelle zur externen Strömungsregelung sowie die Implementierung eines Aktuator-Systems zur Generierung, Überwachung

und Regelung der transversalen Oberflächenwellen in Windkanalexperimenten. Diese hohe Abstraktionsstufe ermöglicht die Aufteilung in einzelne, unabhängige Komponenten in einem modularen Aufbau und stellt die eigenständige Verwendbarkeit einzelner Komponenten sicher.

Das Netzwerk stellt ein Schema für eine mögliche Vernetzung dar. Es wird als kabelgebundenes Netzwerk realisiert und stellt dabei die zentrale Einheit mit Schnittstellen zur Strömungsregelung und zur Aktuator-Ansteuerung dar. Diese Aufteilung wird in Abb. 3.3 verdeutlicht. Die Strömungsregelung befindet sich auf der linken Seite und übergibt an das Netzwerk Daten für die Kontroll-Knoten, Aktuatoren und Sensoren (rote und blaue Elemente). Für die Experimente im Windkanal wird an einem dieser Kontroll-Knoten das Aktuator-System angekoppelt und repräsentiert die Aktuatoren und Sensoren.

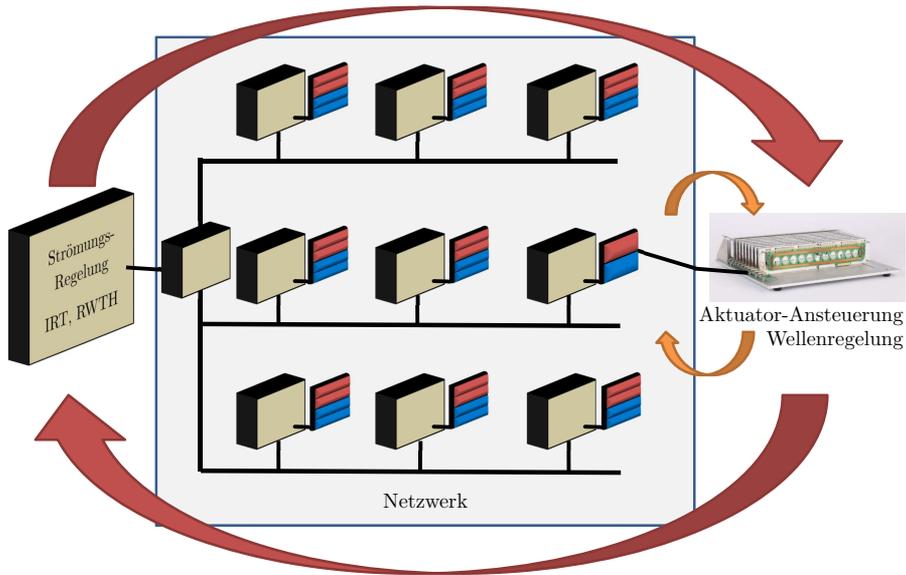
Die Regelgröße für das zeitdiskrete System der Strömungsregelung ist der turbulente Reibungswiderstand. Dieser wird durch das Aktuator-System aktiv beeinflusst. Ausgehend von einer periodischen Aktuierung repräsentieren die folgenden Parameter die Basis für die Stellgröße der Strömungsregelung:

- Amplitude ( $A$ )
- Frequenz ( $F$ )
- Wellenlänge ( $W$ )
- Offset ( $O$ )
- Wellenform mit zusätzlichen Parametern

Die Applikation erfordert eine reibungslose und echtzeitige Übertragung durch das Netzwerk und die zeitnahe Umsetzung der Stellgrößen im Aktuator-System. Außerdem ist die Übertragung von Sensordaten als Messgrößen für die Strömungsregelung notwendig. Dazu sind die Schnittstellen des Netzwerkes sowohl zur externen Regelung als auch zum Aktuator-System eindeutig zu definieren. Für die Verwendung in weiteren Anwendungen muss bei der Entwicklung des Netzwerkes bereits durch den Einsatz flexibler und erweiterbarer Schnittstellen vorgesorgt werden.

Eine denkbare Lösung ist die direkte Übertragung einer Periode des Ansteuerungssignals zur Steuerung der Aktuatorik. Für hohe Abtastraten zur Generierung möglichst glatter Signale und unterschiedliche Ansteuerungsformen erfordert dies jedoch eine sehr hohe Datenrate. Eine Alternative ist die Parametrierung der Signale durch die oben angegebenen Wellenparameter. Durch

eine flexible Gestaltung der Schnittstelle sind Anpassungen für die Übertragung in ein anderes Einsatzgebiet gewährleistet. Die Signalgenerierung zur Aktuator-Ansteuerung, direkt am System, ist weniger aufwändig als die Übertragung einer Periode eines hochaufgelösten Signals. Sie bietet außerdem mehr Flexibilität, zumal am Aktuator-System unterschiedliche Grundformen zur Ansteuerung vorgesehen werden.



**Abb. 3.3:** Das Netzwerk unterteilt die Aufgaben in Kontroll-Knoten sowie Aktuatoren und Sensoren (rote und blaue Elemente). Außerdem sind Schnittstellen für die Strömungsregelung und für die Aktuator-Ansteuerung mit Wellenregelung zur Realisierung eines kaskadierten Regelkreises vorgesehen.

Die externe Regelung benötigt Informationen über die Reaktion des Systems bei einer Änderung der Stellgrößen. Für die Übertragung dieser Reaktion gibt es die gleichen Optionen wie beim Stellsignal. Es können entweder die Rohdaten der Messsignale oder lokal vorverarbeitete Sensordaten zur direkten oder indirekten Bestimmung des turbulenten Reibungswiderstandes für die Strömungsregelung über das Netzwerk übertragen werden. Die Verwendung vorverarbeiteter Sensordaten ermöglicht eine Reduktion des Informationsflusses und eine Zusammenführung mit den gewählten Aktuierungs-Parametern. Die Unterscheidung der Nachrichten erfolgt mittels einer Identifikationsnum-

mer.

Eine flexible Unterteilung des Netzwerkes erlaubt den effizienten Umgang mit inhomogenen Bedingungen. Die entsprechenden Netzwerkbereiche werden mit einem Bereichsparameter gekennzeichnet. Jeweils ein Kontroll-Knoten wird als verantwortlich für einen einzelnen Bereich definiert.

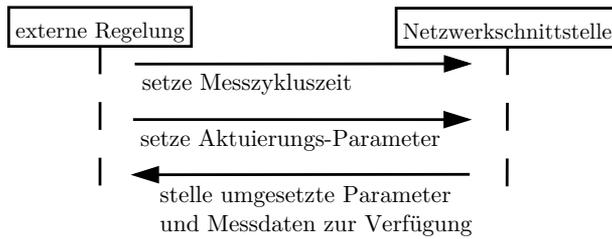
**Tabelle 3.1:** Parameter zur Kommunikation zwischen externer Regelung und der Schnittstelle des Netzwerkes.

Parameter	Verwendung	Datentyp
ID	Identifikationsnummer	integer
<b>error</b>	Fehlerindikator	integer
<b>area</b>	zu aktuierender Bereich	integer
<b>accucontrol</b>	geforderte Genauigkeit der Regelung	double
<b>sampletimesens</b>	Abtastzeit der Sensoren [ <i>ms</i> ]	double
Parameter	Stellgrößen der externen Regelung	diverse

Das vorgesehene Anwendungsgebiet benötigt sowohl allgemeine als auch spezielle Parameter, die über die Schnittstelle zu einer externen Regelung im Netzwerk angenommen werden. Tabelle 3.1 enthält eine Auflistung der bisher definierten und verwendeten Parameter. Es ist vorgesehen, dass die Strömungsregelung sowohl die Stellgrößen als Aktuierungs-Parameter, als auch den Bereich für die Aktuierung (**area**) und die Abtastrate der Sensoren zur Messung des Widerstandes (direkt oder indirekt) vorgibt. Die Fehlerindikation (**error**) ist zur beidseitigen Nutzung vorgesehen.

Der direkt gemessene oder indirekt abgeleitete Reibungswiderstand wird vom Netzwerk an die externe Regelung weitergeleitet. Die Rückmeldung erfolgt zusammen mit der Nachricht über die Anwendung der Parameter und enthält sowohl die umgesetzten Aktuierungs-Parameter, als auch die Größen der Widerstandsmessung für jeden Bereich (siehe Abb. 3.4). Falls alle Bereiche mit einheitlichen Stellgrößen versorgt werden, wird für jeden Bereich eine Rückmeldung gegeben.

Die Beschränkungen der Stellgrößen für die Regelung hängen vom verwendeten Aktuator-System ab. Es muss in jedem Fall sichergestellt werden, dass das Aktuator-System durch die externen Regelvorhaben keinen mechanischen Schaden nimmt. Das Netzwerk muss die übergebenen Parameter entsprechend



**Abb. 3.4:** Die Kommunikation zwischen externer Regelung und Netzwerk sieht die einmalige Übertragung der Messzykluszeit sowie die Übermittlung der Aktuierungs-Parameter vor. Als Antwort werden die umgesetzten Parameter und die Messdaten zusammen in einem Paket erwartet.

auf Plausibilität prüfen und bei einer Verletzung der Beschränkungen eine Rückmeldung geben. Die zugehörigen Kenngrößen sind im Netzwerk hinterlegt.

Für die Realisierung der Oberflächenaktuierung mit dem in Kapitel 4.1 vorgestellten, elektromagnetischen Aktuator-System muss jeder einzelne Aktuator-Balken einer vorgegebenen Trajektorie mit einer vorgegebenen Genauigkeit folgen. Das Aktuator-System benötigt eine große Kraft zur Bewegung der Aluminiumoberfläche bei der Erzeugung der Oberflächenwellen. Inhomogene und instationäre Anströmungsbedingung machen den Einsatz einer Strömungsregelung notwendig. Die Anforderung dieser Regelung an die Echtzeit und auch die räumliche Verteilung von Aktuatoren und Sensoren sind noch nicht bestimmt, da sich die Methoden zur Widerstandsreduktion durch transversale Oberflächenwellen noch in der Entwicklung befinden. Zur Untersuchung dieser Randbedingungen wird das Netzwerk sowohl an die Strömungsregelung als auch an die Aktuatorik angekoppelt. Die Schnittstellen dazu sind in Kapitel 3.3.4 beschrieben.

Eine Änderung der Wellenparameter definiert für die Aktuator-Ansteuerung im inneren Regelkreis zunächst eine Steuerungsaufgabe (siehe Abb. 3.2). Es muss sichergestellt werden, dass keine plötzlichen Übergänge zwischen zwei verschiedenen Wellen zu hochfrequenten Störungen im Strömungsfeld führen. Außerdem müssen kritische mechanische Belastungen des Aktuator-Systems, die aus Parameteränderungen resultieren und zu Beschädigungen führen können, ausgeschlossen werden. Dazu werden in einem späteren Kapitel glatte Signalübergänge an dafür sinnvollen Signalpositionen vorgestellt (siehe Kapi-

tel 4). Zur Erzeugung von wandernden Oberflächenwellen muss die Phasenverschiebung zwischen den Aktuatoren kontrolliert werden.

Die Genauigkeit der Ansteuerung wird durch die externe Regelung vorgegeben. Zunächst wird eine Abweichung von maximal fünf Prozent gegenüber dem Ansteuerungssignal, also der gewünschten Trajektorie, in der Aktuatorregelung als annehmbar angenommen. Für ein sinusförmiges Ansteuerungssignal wird ein zeitbasiertes Ausgangssignal in Gleichung 3.1 definiert.

$$S(t) = A \cdot \sin(F \cdot 2\pi \cdot t + P) + O \quad (3.1)$$

Die Phasenverschiebung  $P$  relativ zum Nachbarbalken ergibt sich mit dem Balkenabstand  $d$  direkt aus der Wellenlänge  $W$ .

$$P = \frac{d \cdot 2\pi}{W} \quad (3.2)$$

Die Kopplung der einzelnen Aktuator-Balken über die Aluminiumoberfläche führt zu einer gegenseitigen Beeinflussung der Balkenbewegung (siehe Kapitel 4.3). Mithilfe einer Wellenregelung wird die geforderte Genauigkeit der Oberflächenaktuierung sichergestellt und mögliche Störgrößen werden ausgeglichen, wie beispielsweise Einflüsse der Luftströmung auf die Oberfläche. Die Wellenregelung bildet somit den zweiten Teil des inneren Regelkreises und wird zusammen mit der Aktuator-Ansteuerung realisiert. Das Ziel dabei ist die Minimierung der Abweichung zwischen Oberflächenbewegung und der vorgegebenen Trajektorie der einzelnen Aktuatoren. Sie ist in Kapitel 4.4 beschrieben.

## 3.2 Validierungsstrategie

Das Netzwerkkonzept wird mithilfe eines Modells in einer Simulation validiert. Zunächst wird das Netzwerk selbst auf Plausibilität geprüft, bevor es in die Anwendung integriert wird.

Das Verhalten des Netzwerkes während der Simulation muss mit dem in der realen Welt übereinstimmen. Zu diesem Zweck werden grundlegende Parameter wie Bandbreiten und Protokoll-Übertragungszeiten innerhalb einer Testumgebung evaluiert. Bei der Verwendung eines identischen Kommunikationsparadigmas sowohl in der Simulation als auch in der Testumgebung

darf eine maximale Abweichung von einer Millisekunde zwischen Kommunikationsereignissen innerhalb eines Kommunikationszyklus nicht überschritten werden. Damit lassen sich falsche Annahmen vermeiden, die bei der späteren Hochskalierung der Netzwerke zu sehr großen Fehlern führen können.

Die Anforderungen an das Netzwerk sind die echtzeitige Übertragung der vorgegebenen Stellgrößen der externen Regelung zum Aktuator-System sowie die Rückmeldung mit entsprechenden Messgrößen in Echtzeit. Dazu muss die echtzeitige Übertragung innerhalb des Netzwerkmodells sichergestellt werden. Vorgegebene Beschränkungen dürfen nicht verletzt werden, sodass die entsprechenden Datenpakete gesendet und die zugehörige Antwort innerhalb eines definierten Kommunikationszyklus bereitgestellt werden.

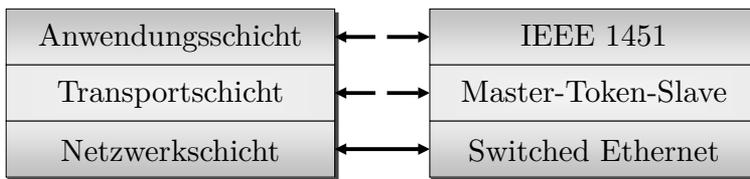
Ist das korrekte Verhalten der Kommunikation sichergestellt, so erfolgt mithilfe der externen Reglerkomponente die Validierung des Gesamtmodells. Die Stellgrößen werden für die Anwendung in homogenen und inhomogenen Randbedingungen unterschieden, sodass entweder eine einzelne Nachricht für alle Bereiche, oder für jeden Bereich eine separate Nachricht zuzustellen ist. Es muss sichergestellt werden, dass von jedem Bereich die Messdaten zusammen mit den umgesetzten Aktuierungs-Parametern an die externe Regelung übergeben werden. Diese Funktionalität wird validiert, indem die ausgehenden und eingehenden Daten aus der Sicht der externen Regelung miteinander verglichen werden. Dabei muss sowohl der Inhalt auf Korrektheit, als auch die Einhaltung des zeitlichen Rahmens geprüft werden. Kommunikationsfehler, die nicht durch induzierte Fehler entstehen, zeigen, dass eine Verletzung der zeitlichen Beschränkung innerhalb der Kommunikation auftritt.

Ein validiertes Netzwerk ermöglicht die Ankopplung und den Nachweis der Funktionalität der externen Strömungsregelung und der Aktuatorik mithilfe einer Model-in-the-Loop-Simulation. Sofern die Stellgrößen und entsprechenden Messdaten durch die Schnittstellen zwischen Strömungsregelung und Aktuator-System übertragen werden, ist die spezifizizierte Funktionsweise nachgewiesen.

Die korrekte Anwendung und Änderung der Aktuierungs-Parameter lässt sich durch Referenzmessungen sowohl am analogen Ausgang als auch mithilfe des Aktuierungs-Systems bestätigen.

## 3.3 Netzwerkschichten gemäß OSI-Referenzmodell

Für das nun folgende Konzept wird eine Verallgemeinerung der Anwendung vorgenommen. Die „externe Regelung“ bezeichnet dabei im speziellen Anwendungsfall die Strömungsregelung und das „Aktuator-System“ repräsentiert die oszillierende Oberfläche zur Beeinflussung des turbulenten Reibungswiderstandes.



**Abb. 3.5:** Der Protokollstapel zur Kommunikation innerhalb des Netzwerkes ist auf drei Ebenen reduziert und lässt sich durch ein Protokoll je Schicht spezifizieren.

Die Kommunikation wird in drei Ebenen realisiert, analog zu den in Kapitel 2.1 vorgestellten Referenzmodellen. Im Gegensatz zu den dort beschriebenen Ansätzen mit zwei Schichten im industriellen oder Automotive Umfeld wird ein zusätzliches Protokoll eingeführt. Deleuze [23] erklärt bereits, dass sich das Internet-Referenzmodell auf drei Ebenen reduzieren lässt. In der vorliegenden Anwendung wird das Kommunikationsmodell in Applikationsschicht, Transportschicht und Netzwerkschicht unterteilt (siehe Abb. 3.5). Es wird eine einfache Aufgabenverteilung sichergestellt. Dabei werden die verfügbaren Schichten des OSI-Referenzmodells teilweise zusammengefasst. So wird sowohl ein zusätzlicher Overhead als auch die Implementierung nicht notwendiger Protokolleigenschaften vermieden.

### 3.3.1 Anwendungsschicht

Die beiden oberen Schichten des Referenzmodells werden zur Anwendungsschicht zusammengefasst und sorgen für die einheitliche Datendarstellung und die Bereitstellung der Kommunikation als Schnittstelle zur Anwendung. Diese Schicht wird analog zum IEEE 1451 *Smart Transducer Interface Standard*

aufgesetzt. Dieser Standard wurde zur Vernetzung von Aktuatoren und Sensoren entwickelt und ist in Kapitel 2.1.4 beschrieben. Das Protokoll bietet eine vielseitige, standardisierte Kommunikationsschnittstelle, die eine flexible Anbindung der Knoten ermöglicht.

Eine alternative Möglichkeit bietet das Anwendungsprotokoll *Real-Time Transport Protocol* (RTP) [77]. Dieses arbeitet oberhalb der Transportschicht innerhalb des Internet-Protokollstapels (siehe Kapitel 2.1.2). Durch die Verwendung des UDP-Protokolls wird jedoch ein zusätzlicher Overhead erzeugt. Außerdem ist die Kompatibilität zu einem internetbasierten Netzwerk für das vorgestellte Netzwerk nicht notwendig.

**Tabelle 3.2:** Beschreibung der Knoten gemäß den Bezeichnungen im IEEE 1451 Standard.

Name	Handlungsweise	Aufgabe
STM <sub>S</sub>	Sensor-Knoten	Aufnahme, Verarbeitung und Bereitstellung von Messdaten
STM <sub>A</sub>	Aktuator-Knoten	Steuerung und Regelung
NCAP <sub>C</sub>	Kontroll-Knoten	Empfang und Verteilung der Daten des NCAP <sub>I</sub> Empfang und Zusammenfassung der Daten der STM zur Weiterleitung an den NCAP <sub>I</sub>
NCAP <sub>I</sub>	Schnittstellen-Knoten	Kontroll-Knoten zur Bereitstellung jeweils einer Schnittstelle zur externen Regelung

Die Benennung der einzelnen Knoten erfolgt in Anlehnung an den IEEE 1451 Protokollstandard. Dabei werden die bereits eingeführten Aktuator-, Sensor- und Kontroll-Knoten entsprechend ihrer Aufgaben eingeordnet. Die Gruppe der Kontroll-Knoten wird mit NCAP (*Network Capable Application Processor*) bezeichnet, die Gruppe der Aktuator- und Sensor-Knoten jeweils mit STM (*Smart Transducer Module*). Jede Knotengruppe bekommt eine Aufgabe zugeteilt, dabei wird in der Gruppe der Kontroll-Knoten im weiteren Verlauf zwischen NCAP<sub>I</sub>, der Gruppe der Schnittstellen-Knoten, und NCAP<sub>C</sub>,

den inneren Kontroll-Knoten, unterschieden. Die Gruppe der STM wird in  $STM_A$ , die Aktuator-Knoten, und  $STM_S$ , die Sensor-Knoten unterteilt. Unabhängig von der physikalischen Trennung von Intelligenz und Hardware-Anbindung bezeichnet *Smart Transducer Module* den Verbund von NCAP und TIM (*Transducer Interface Module*) analog zu einem beliebigen Smart Transducer Modell [79].

Zum einfacheren Verständnis werden im Folgenden diese Abkürzungen beibehalten (siehe Tabelle 3.2). Zur Identifikation der Position beziehungsweise Zugehörigkeit werden die Knoten nummeriert. Die Sicherstellung der Echtzeitfähigkeit erfolgt auf einer der unteren Schichten, da der Aufwand für die zeitliche Abstimmung der Aufgaben der einzelnen Knoten auf der Anwendungsschicht sehr hoch ist.

#### 3.3.2 Transportschicht

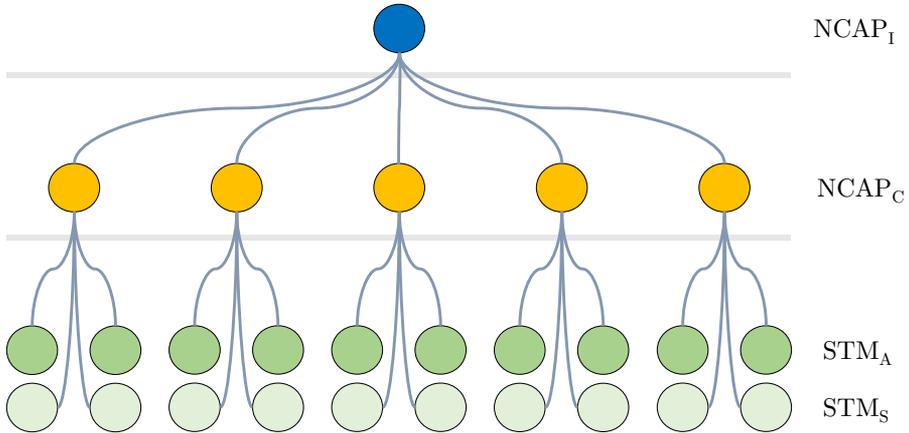
Die Transportschicht fasst die Ebenen drei, vier und fünf des OSI-Referenzmodells zusammen. Hier wird die Paketvermittlung, die Synchronisation sowie die Datenübertragung zwischen den Knoten realisiert. Wird die Nachrichtengröße auf die *Maximum Transfer Unit* (MTU) des MAC-Protokolls beschränkt, so müssen die spezifischen Funktionen Segmentierung, Flusskontrolle und Staukontrolle nicht weiter betrachtet werden.

Die Einführung eines echtzeitfähigen Protokolls auf der Transportschicht vereinfacht die Beschreibung des Systems. In Anlehnung an die beschriebenen middlewarebasierten Protokolle in Kapitel 2.2.2 wird das selbstentwickelte *Master-Token-Slave-Protokoll* (MTS) als Übertragungsprotokoll auf der Transportschicht verwendet [28]. Das Protokoll teilt die Kommunikation mithilfe einer Master-Slave Strategie in unterschiedliche Abschnitte und verwendet ein deterministisches Send- und Empfangsverfahren, um die Datenübertragung zu einer vorgegebenen Zeit, also in Echtzeit, sicherzustellen.

#### 3.3.3 Netzwerkschicht

Die Netzwerkschicht fasst die unteren beiden Ebenen des OSI-Referenzmodells zusammen. Dabei wird das MAC-Protokoll ausgewählt und die Bitübertragung auf dem entsprechenden Kommunikationsmedium festgelegt. Letztere ist für die theoretische Betrachtung eines Netzwerkes

nicht relevant. An dieser Stelle wird auch die Vernetzungsstrategie beschrieben. Entsprechend der Anforderungen wird ein kabelgebundenes Netzwerk entwickelt.



**Abb. 3.6:** Die Baumstruktur stellt die virtuelle Aufteilung und Vernetzung der verschiedenen Knotentypen dar. Das Verhältnis von  $\text{NCAP}_C$  zu  $\text{NCAP}_I$  und  $\text{STM}$  zu  $\text{NCAP}_C$  kann dabei angepasst werden. Die virtuelle Aufteilung spiegelt nicht die räumliche Verteilung der Knoten in der Anwendung wider.

Aus der Definition des *Master-Token-Slave*-Protokolls für die Transportschicht und den bisherigen Vorüberlegungen lässt sich eine Hierarchie zur Verteilung der Knoten ableiten. In dieser Baumstruktur dient der  $\text{NCAP}_I$ -Knoten als Master der  $\text{NCAP}_C$ . Die  $\text{NCAP}_C$  sind jeweils Master der  $\text{STM}$ -Knoten. Die entsprechende Vernetzung ist in Abb. 3.6 dargestellt. Jeder  $\text{NCAP}_C$ -Knoten der mittleren Ebene verwaltet einen separaten Bereich und lässt sich über den beschriebenen *area*-Parameter durch die externe Regelung ansprechen (siehe Abb. 3.6). Zur Vereinfachung der Darstellung wird im Folgenden nur ein  $\text{NCAP}_I$  angenommen und das Netzwerk analog zu dieser Struktur entwickelt. Neben einer eventuellen Redundanz oder einer Vergrößerung der Knotenanzahl bietet die Betrachtung mehrerer  $\text{NCAP}_I$  im Folgenden keine entscheidenden Unterschiede oder Einschränkungen. Die Modellierung in einer Baumstruktur ermöglicht parallele Abläufe, spart also Zeit und erhöht die Effizienz. Für die entsprechenden Anwendungen werden homogene Bedingungen für eine Gruppe von  $\text{STM}_A$  in einem Bereich als Annahme zugrunde gelegt. Die maximale Anzahl von  $\text{STM}_A$ , die einem  $\text{NCAP}_C$  zugeordnet werden, stellt

eine weitere Randbedingung dar.

Für die MAC-Schicht stehen verschiedene Protokolle zur Verfügung. Mögliche Protokolle wie Ethernet, CAN oder FlexRay sind in Kapitel 2.1 beschrieben. Für den Einsatz im großflächigen Aktuator- und Sensornetzwerk wird ein switched Ethernet Protokoll verwendet. Hardwarekomponenten zum Aufbau einer Testumgebung mit minimaler Teilnehmerzahl lassen sich durch ein sternförmiges Netzwerk mithilfe eines Switches einfach und schnell verbinden. Die gewonnenen Erkenntnisse für switched Ethernet basierte Netzwerke lassen sich mit geringen Einschränkungen auf ein Ethernet Bussystem übertragen.

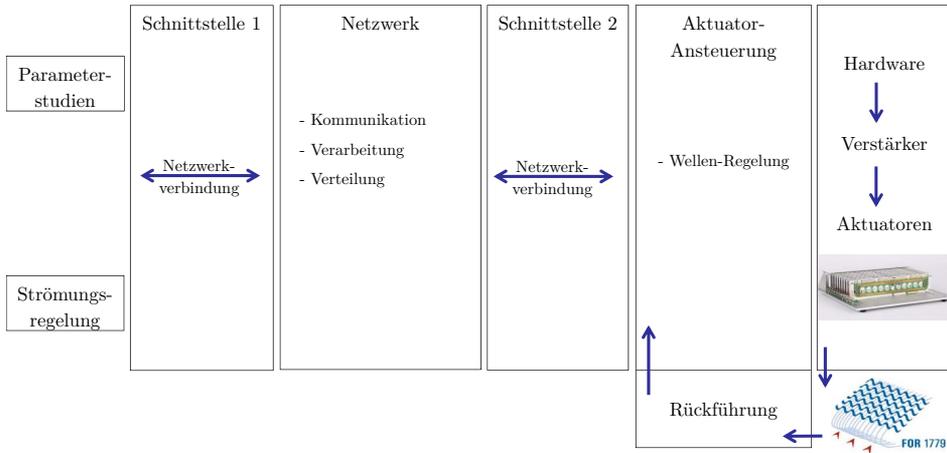
Die Umsetzung in einem einzelnen Netzwerk ist bei der hohen Anzahl von Knoten nicht möglich. In einem kurzen Beispiel lässt sich dies veranschaulichen: Ein Netzwerk mit zehn Teilnehmern, die jeweils in einem Zyklus von zehn Millisekunden ein Paket senden, ist bei einer Übertragungszeit von einer Millisekunde pro Paket bereits ausgelastet und kann bei geringen Verzögerungen – beispielsweise durch Kollisionen – die Übertragung der Pakete nicht weiter garantieren. Es gibt dann die Möglichkeit den Sende-Zyklus zu erhöhen oder die Anzahl der Teilnehmer zu begrenzen. Da der benötigte Sende-Zyklus noch unbekannt ist, wird die Möglichkeit zur Begrenzung der Netzwerkteilnehmer mit in das Netzwerk aufgenommen.

#### 3.3.4 Externe Netzwerkschnittstellen

Zur einfacheren Entwicklung der Module zur Strömungsregelung beziehungsweise zur Ankopplung einer externen Regelung an die Netzwerksimulation und das Aktuator-System müssen zunächst die Schnittstellen zwischen den einzelnen Komponenten definiert werden. Diese Schnittstellen ermöglichen eine eigenständige Entwicklung der Bestandteile. Dabei wird auf etablierte Lösungen, die auch langfristig verfügbar sein werden, zurückgegriffen.

Für das Konzept werden die notwendigen Module und deren Schnittstellen unterteilt. Ausgangspunkt dazu sind die Anforderungen sowohl der Parameterstudien im Windkanal, als auch der Strömungsregelung (siehe Abb. 3.7).

Die Schnittstellen zur externen Strömungsregelung und zur Aktuator-Ansteuerung werden als Netzwerkverbindung definiert. Die entsprechenden Aktuierungs-Parameter werden an das Netzwerk übergeben und verteilt. Die direkte Netzwerkverbindung bietet eine einfache und flexible Vernetzung der Systeme.



**Abb. 3.7:** Das Grundkonzept definiert die Schnittstelle zwischen den Parameterstudien beziehungsweise der externen Strömungsregelung und dem Netzwerk sowie die Schnittstelle des Netzwerkes zur Aktuator-Ansteuerung mit integrierter Datenrückführung zur Schließung des Regelkreises.

### 3.4 Modellierung des Aktuator- und Sensornetzwerkes

Das Netzwerk wird zur flexiblen Anpassung der räumlichen Verteilung und zur Berücksichtigung der Randbedingungen der externen Regelung innerhalb eines Modells implementiert. Das Aktuator- und Sensornetzwerkmodell dient zur Untersuchung der Topologie und Knotenverteilung. Eine günstige Anzahl der Knoten aus jeder Gruppe ( $NCAP_C$ ,  $STM_A$ ,  $STM_S$ ) ist zu bestimmen. Die Vernetzungsstrategie ist zu untersuchen und eine möglichst plausible Lösung für die Anwendung in der Strömungsregelung ist zu bestimmen. Die Entwicklung des Netzwerkmodells und die Simulation spezieller Kommunikationsabläufe liefern ein besseres Verständnis für die notwendige Konfiguration des vorgesehenen physikalischen mikrocontrollerbasierten Netzwerkes. Mögliche Fehlerquellen können im Modell identifiziert und so in der physikalischen Umsetzung vermieden werden. Dabei wird die Netzwerkkommunikation in drei Schichten: Netzwerkschicht, Transportschicht und Anwendungsschicht modelliert. Das Netzwerk wird im weiteren Verlauf als geschlossene Einheit

ohne äußere Einflüsse betrachtet.

Die Grundlage für das Netzwerkmodell und für die Implementierung der Netzwerkschicht bildet die TrueTime-Toolbox (siehe Kapitel 2.4). Auf der Transportschicht wird das echtzeitfähige *MTS*-Protokoll modelliert und implementiert. Das Modell der Anwendungsschicht wird über grundlegende und notwendige Funktionalitäten des IEEE 1451.1 als Substandard des IEEE 1451 beschrieben. Das Verhalten der einzelnen Knoten wird mithilfe von Zustandsautomaten modelliert und in jeden einzelnen Knoten implementiert. Wichtig dabei ist die genaue Abstimmung der Komponenten aufeinander, um die effiziente Datenübertragung zu gewährleisten.

### 3.4.1 Werkzeuge und Implementierung

Es wird eine generische Lösung zur effizienten Erzeugung und Simulation von Aktuator- und Sensornetzwerkmodellen implementiert, die den bisher definierten Anforderungen genügen. Dabei wird die bisher festgelegte Aufgabenverteilung in die drei Gruppen Kontroll-, Aktuator- und Sensorknoten berücksichtigt. So kann flexibel auf zusätzliche Anforderungen der Applikation eingegangen werden.

Sowohl die protokollbasierten Parameter als auch die Definition von Abläufen als Reaktion auf eintretende Ereignisse bestimmen das Verhalten des Modells in der Simulation. Die Modellcharakterisierung erfolgt durch zeitliche Beschränkungen, dabei werden Übertragungszeiten und Rechenzeiten in den verschiedenen Ebenen eingeführt.

Für die Modellierung und Simulation des Netzwerkes sowie die Bereitstellung der Schnittstellen zur externen Strömungsregelung und zur Aktuator-Ansteuerung wird MATLAB/Simulink sowie die TrueTime-Toolbox verwendet. Die Modellierung des Netzwerkes erfolgt mithilfe eines ereignisbasierten Ansatzes. Die Netzwerkteilnehmer warten auf eingehende Daten und entscheiden anhand dieser, welche Aktion auszuführen ist.

Die TrueTime-Toolbox bietet die Möglichkeit, eingebettete Systeme und Netzwerkverbindungen mittels Kernel- und Netzwerk-Blöcken in deterministischen Simulationen zu implementieren. Eine detaillierte Beschreibung der Funktionen, die die TrueTime-Toolbox zur Verfügung stellt, ist im Handbuch [17] zu finden. Im Folgenden wird der Einsatz einiger der Funktionen beschrieben. Innerhalb der Kernel-Blöcke werden sogenannte *Code-Funktionen* zur Realisierung verwendet. Es sind verschiedene Zeitverteilungsrichtlinien für die Prozesse der Knoten in TrueTime auf Quellcode-Ebene einstellbar:

- fixed-priority (`prioFP`)
- deadline-monotonic (`prioEDF`)
- earliest-deadline-first (`prioDM`)

Es gibt aber auch die Möglichkeit eigene Richtlinien zu verfassen. Im beschriebenen Modell werden alle Prozesse gleichberechtigt behandelt (`fixed-priority`). Analoge Ein- und Ausgänge der Kernel-Blöcke ermöglichen die Anbindung an Simulink-Subsysteme zur Signalübermittlung (`ttAnalogIn`, `ttAnalogOut`). Innerhalb der Implementierung des *MTS*-Protokolls und der Realisierung der Applikationsfunktionalität wird sowohl die `ttSleep`-Methode, als auch der Rückgabewert für die Simulation der Rechenzeit [17] verwendet. Damit lassen sich nebenläufige Prozesse innerhalb der Kernel-Blöcke umsetzen.

Der Datenaustausch auf der Netzwerkschicht erfolgt über die `ttSendMsg` und `ttGetMsg`-Methoden. Bei der Versendung der Daten wird anhand der vorgegebenen Parameter im Netzwerk-Block die Übertragungszeit berechnet und zusätzliche Randbedingungen, wie Hardwarebeschränkungen werden berücksichtigt [17]. Es ist von relativ kurzen Leitungen auszugehen, sodass keine gravierenden Laufzeitunterschiede in der Paketübermittlung angenommen werden. `TrueTime` stellt jedoch nur das MAC-Protokoll und die emulierte Übertragung der Daten in der Simulation zur Verfügung. In Anlehnung an das Schichtenmodell müssen die weiteren Schichten, Transportschicht und Anwendungsschicht neben der eigentlichen Funktionalität jedes Knoten implementiert werden.

Auf der Transportschicht wird das echtzeitige *MTS*-Protokoll verwendet. Die Kommunikation wird mit einer Master-Slave-Strategie in Kommunikationszyklen aufgeteilt. Das Protokoll wird mithilfe der Code-Funktionen implementiert. Zur Vereinfachung wird die Kommunikationsschnittstelle der Anwendungsschicht mit der Anwendung zusammengefasst und das IEEE 1451.1 Protokoll durch Datendarstellung und Übermittlungszeiten repräsentiert. Für die einzelnen Kommunikationsphasen des *MTS*-Protokolls sowie für den Nachrichtenempfang und die Anwendungsebene ist jeweils eine Code-Funktion implementiert. Zusätzlich existieren Funktionen zur Bereitstellung der Schnittstellen für die Transportschicht. Zur einfachen Modifikation des Modells werden der Quellcode für die Simulation und die globalen und variablen Modellparameter strikt getrennt. Es wird ein Werkzeug zur einfachen Erzeugung großflächiger Netzwerkmodelle in MATLAB/Simulink implementiert.

Für die experimentelle Validierung der Strömungsregelung wird das Netzwerkmodell an die externe Regelung und das Aktuator-System innerhalb einer Model-in-the-Loop-Simulation angebunden. Der Betrieb des Aktuator-Systems wird durch eine LabVIEW-Real-Time-basierte Software gesteuert, die eine echtzeitige Umsetzung der Stellgrößenänderungen ermöglicht.

Für Modellimplementierung und Ansteuerung der Aktuatoren werden die Vorteile von MATLAB/Simulink und LabVIEW ausgenutzt. MATLAB/Simulink als flexibles und mächtiges Datenverarbeitungssystem für die Modellierung und Simulation des Netzwerkes und LabVIEW für die flexible und echtzeitige Ansteuerung der Aktuatorik und das Auslesen der Sensorik (siehe Kapitel 2.4)

### 3.4.2 Modellierung der Netzwerkschicht

Die bereits beschriebene Baumstruktur berücksichtigt die Konfiguration entweder als gemeinsames, gruppiertes oder vollvermaschtes Netzwerk, wobei im letzten Fall jeder Knoten eine Verbindung mit jedem Knoten besitzt. Im Folgenden erfolgt die Aufteilung gruppiert, sodass die Verbindung zwischen  $NCAP_I$  und den  $NCAP_C$  in einem abgeschlossenen Netzwerk erfolgt, genauso wird die Verbindungen der einzelnen  $NCAP_C$  mit ihren zugeordneten STM-Knoten etabliert. Somit benötigt jeder  $NCAP_C$  zwei physikalische Netzwerkverbindungen. Diese Aufteilung bietet eine hohe Effizienz, da der Nachrichtenaustausch und die Datenverarbeitung in mehreren Bereichen parallel abgewickelt wird.

Im Modell wird ein switched Ethernet zur Realisierung eines Stern-Netzwerkes implementiert. Die Nachrichten werden entweder mit einer bestimmten Adresse oder als Broadcast versendet. Zur vereinfachten Verwendung des MAC-Protokolls werden in der Simulation ausschließliche Broadcasts zur Datenübermittlung verwendet und die Adressierung wird auf die Transportschicht verlagert.

Für die Modellierung der Netzwerkschicht in TrueTime werden protokollspezifische Parameter des verwendeten Kommunikationsprotokolls benötigt. Beispielsweise muss die Puffergröße des Switches bei einem switched Ethernet Netzwerk beschrieben werden. Im Folgenden werden die nötigen Parameter aufgelistet, wobei die Puffergröße zu den Hardwarebeschränkungen zählt:

- Bandbreite
- minimale / maximale Paketgröße

- Wahrscheinlichkeit für Paketverluste
- Hardwarebeschränkungen

Variable, für die Code-Funktionen relevante Parameter werden für die einfache Änderbarkeit in eine Konfigurationsdatei ausgelagert. Jeder Knoten greift in der Simulation auf diese Datei zu. So ist sichergestellt, dass alle Knoten die gleichen Informationen erhalten und die gleiche Basis zugrunde liegt.

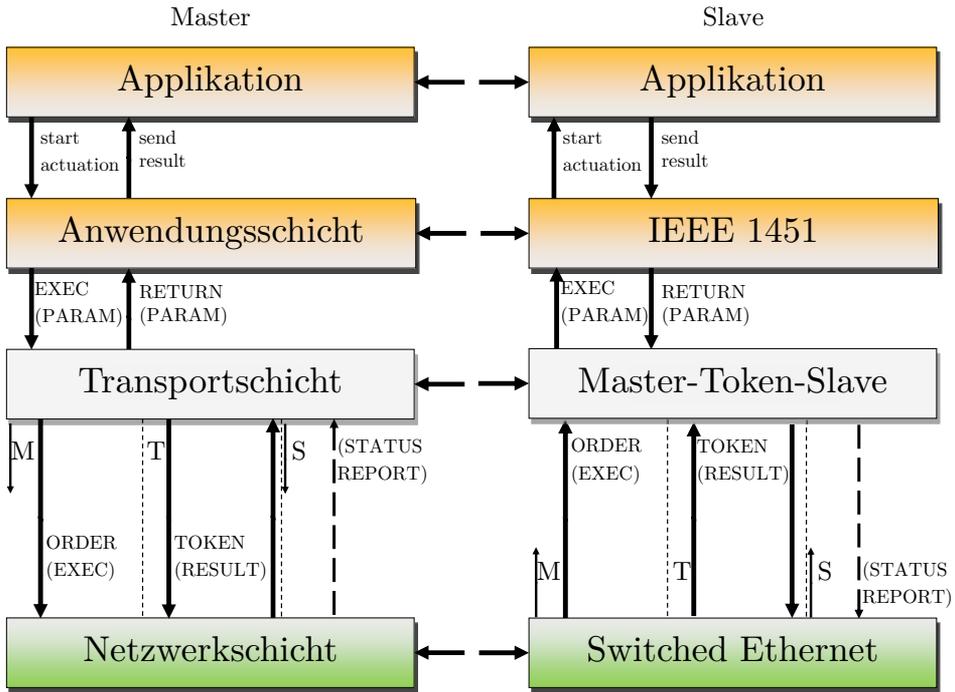
## Fehlerbehandlung

Die Baumstruktur lässt sich, sofern notwendig, durch zusätzliche Verbindungen in ein vollvermaschtes Netzwerk mit Verbindungen zwischen allen Knoten übertragen, um zusätzliche Redundanzen zu erhalten. Bei der Überführung in ein vollvermaschtes Netzwerke ist darauf zu achten, dass weiterhin parallele Kommunikation in den Netzwerken ermöglicht wird um Fehler auf höheren Schichten auszugleichen. Bei der Verwendung von switched Ethernet werden Paketverluste nicht innerhalb des MAC-Protokolls Korrigiert, sodass eine erneute Versendung durch Protokolle auf anderen Kommunikationsschichten initiiert werden muss.

### 3.4.3 Modellierung der Transportschicht

Die Transportschicht ist für die Abstimmung des Datenaustauschs im Netzwerk zuständig. Insbesondere ist die zeitliche Koordination der Abläufe und Abstimmung der Komponenten aufeinander wichtig, um die Echtzeitigkeit zu realisieren. Dazu wird das *Master-Token-Slave*-Protokoll mit einer geordneten Kommunikation implementiert. Das Kommunikationsprinzip über die verschiedenen Netzwerkschichten ist in Abb. 3.8 dargestellt. Die Kontrollnachrichten auf der Ebene der Transportschicht werden mit der minimalen Paketgröße des Ethernets von 64 Bytes versendet.

Mithilfe der Aufteilung des Netzwerkes in Master- und Slave-Knoten wird der deterministische Datenaustausch realisiert. Die drei Kommunikationsphasen (Master, Token, Slave) sind flexibel anpassbar. Der Master gibt den vollständigen Kommunikationsablauf vor. Die Nachrichten werden bis zur Versendung in Warteschlangen vorgehalten. Die Kommunikationsphasen werden durch sogenannte Beacons, also kleinen Datenpaketen, angelehnt an den WLAN Standard [37], initiiert. Die Master- und die Slave-Phase werden mit einem Beacon-Broadcast ausgehend vom Master eingeleitet. Es ist kein Beacon zur



**Abb. 3.8:** Das detaillierte drei Ebenen Modell der Kommunikation regelt den Datenaustausch zwischen den vertikalen Ebenen. Die Aufteilung der Transportschicht in drei Kommunikationsphasen ermöglicht eine deterministische Datenübertragung.

Indikation der Token-Phase notwendig, da der Master die Slaves in der Token-Phase steuert. Steuernachrichten, wie Beacons sind zur internen Kommunikation in der Transportschicht vorgesehen und werden der Anwendungsschicht nicht mitgeteilt.

Die Nachrichten im *MTS*-Rahmen (siehe Auflistung 3.1) besitzen eine eindeutige Identifikationsnummer, mit der sie identifiziert werden. Steuernachrichten auf der Transportebene sind ihren entsprechenden Anfragen durch die Identifikationsnummer zugeordnet.

Multicasts zu  $STM_A$  und  $STM_S$  werden über Broadcasts auf der Netzwerkschicht und mithilfe der Adresse des *MTS*-Rahmens (siehe Auflistung 3.1) realisiert. Pakete, die nicht für den jeweiligen Knoten-Typ bestimmt sind, werden verworfen.

**Auflistung 3.1:** Der *MTS*-Paketrahmen wird vom Transportprotokoll generiert und besteht aus der Quell- und der Zieladresse sowie einer Identifikationsnummer. Die Zugehörigkeit zu einer bestimmten Kommunikationsphase wird genau wie der Nachrichtentyp mitgesendet. Außerdem ist ein Zeitstempel für die Synchronisation vorgesehen. Der eigentliche Datensatz aus der Anwendungsschicht wird in der Variablen *data* gespeichert.

```
enum id {BEACON, MASTER, TOKEN, SLAVE};
2 enum type {MSG, TOK, ACK, ERR, B_M, B_T, B_S};

4 struct mts_msg
  {
6     int from;
      int to;
8     int id;
      // message phase
10    enum id ident;
      // message type
12    enum type typ;
      double timestamp;
14    // data
      struct I1451_msg data;
16  };
```

Das Protokoll wird durch mehrere Tasks in TrueTime modelliert und verwendet die TrueTime-Schnittstellen für die Kommunikation über das MAC-

Protokoll. Die Tasks warten oder sind blockiert. Definierte Schnittstellenfunktionen ermöglichen die Verwendung des Protokolls in der Anwendungsschicht.

### **Master-Phase**

In der ersten Phase, der Master-Phase, hat der Master-Knoten die Möglichkeit, Nachrichten an seine Slaves zu senden. Die Warteschlange, in der die Nachrichten aus der Anwendungsschicht übertragen werden, wird Nachricht für Nachricht abgearbeitet. Die zu sendende Nachricht wird dabei in den Paketrahmen eingefügt und die entsprechenden zusätzlichen Parameter gesetzt.

### **Token-Phase**

In der Token-Phase erhalten alle Slaves, die in der Master-Phase eine Nachricht erhalten haben, die Aufforderung, eine Antwort zu dieser Nachricht zu senden. Diese Aufforderung wird hier als „Token“ bezeichnet und wird nicht als Broadcast oder Multicast versendet, sondern direkt adressiert. Der Slave-Knoten sendet die entsprechende Antwort zu der Identifikationsnummer in der erhaltenen Nachricht, sofern sie in seinem Nachrichtenpuffer vorhanden ist. Andernfalls wird entweder keine Antwort gesendet, sofern sich die zugehörige Nachricht noch im Eingangspuffer befindet, oder es wird ein Fehler zurückgegeben, der anzeigt, dass die entsprechende Nachricht (Eingang oder Ausgang) verloren gegangen ist.

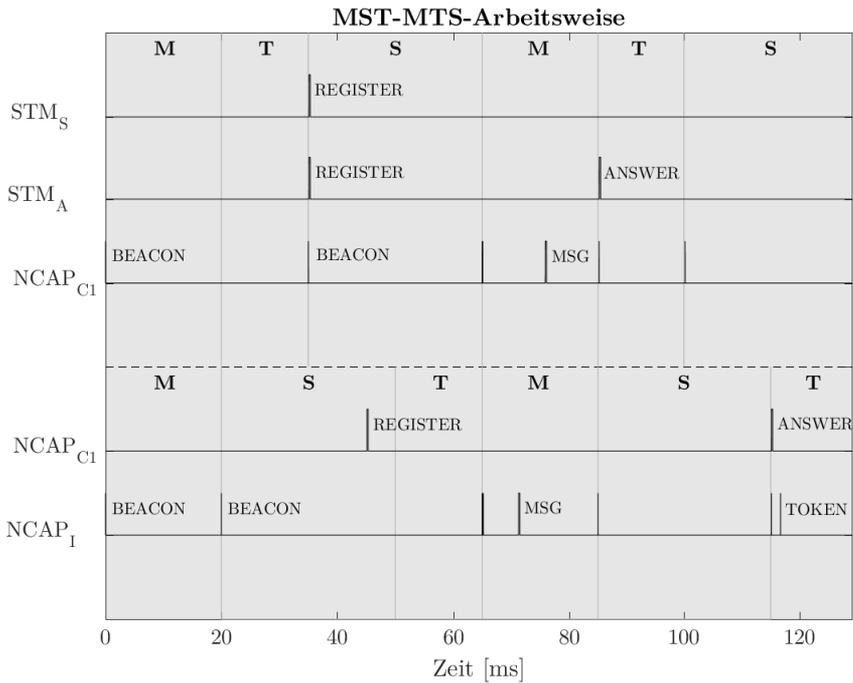
### **Slave-Phase**

Die Slave-Phase stellt den Slave-Knoten die Möglichkeit zur Verfügung, etwaige Fehler mitzuteilen. Diese Nachrichten werden durch den Master mit einem Acknowledgement in der darauffolgenden Master-Phase bestätigt.

### **Anwendung des *MTS*-Protokolls in der Netzwerksimulation**

Die Aufteilung des Netzwerkes in einer Baumstruktur benötigt eine Kommunikation in zwei Ebenen, sodass das *MTS*-Protokoll mehrfach in einer geschachtelten Kommunikation zum Einsatz kommt. Die Anwendung dieses

Protokolls parallel in beiden Ebenen (NCAP<sub>I</sub>-NCAP<sub>C</sub> und NCAP<sub>C</sub>-STM) benötigt ausgedehnte Zykluszeiten, um Pakete vom Schnittstellenknoten durch das Netzwerk zu ihrem Bestimmungsort (STM) zu übertragen. Für die obere Ebene (NCAP<sub>I</sub>-NCAP<sub>C</sub>) bietet sich ein Austausch der Token- und Slave-Phasen an, sodass bei der Kommunikation zwischen NCAP<sub>I</sub> und NCAP<sub>C</sub> ein Zeitfenster für die Verarbeitung und den Austausch zwischen NCAP<sub>C</sub> und STM in der Slave-Phase bereitsteht. Somit erhält das Aktuator- und Sensornetzwerk auf der Transportschicht eine *MST-MTS*-Struktur.



**Abb. 3.9:** Der TrueTime-Graph verdeutlicht die Aufteilung in Kommunikationsphasen sowie die *MST-MTS*-Funktionsweise. Im minimalen Beispiel wird sowohl der Registrierungsprozess als auch ein einzelner Sendezyklus durchgeführt. Das zweite Token ausgehend vom NCAP<sub>I</sub> ist für den nicht dargestellten NCAP<sub>C2</sub> vorgesehen.

Mithilfe der Simulation verschiedener Konfigurationen ist ein funktionierendes Setup zu erarbeiten und Zeitfenster für die jeweiligen Kommunikationsphasen in den verschiedenen Netzwerkebenen der Baumstruktur sind zu bestimm-

men.

Die Abb. 3.9 verdeutlicht die Verschachtelung der beiden Netzwerke in einem TrueTime-Plot (siehe Kapitel 2.4). Beide Zykluszeiten werden mit der gleichen Länge konfiguriert. Zunächst beginnt dem Protokoll entsprechend jeder der Master-Knoten die Beacons per Broadcast an seine Slaves zu verteilen. In der ersten Slave-Phase melden sich alle  $STM_A$  und  $STM_S$  bei ihrem entsprechenden  $NCAP_C$ -Master mit jeweils einem Paket an. Nach Abschluss dieses Vorgangs melden sich die  $NCAP_C$  bei dem  $NCAP_I$  auf dieselbe Weise an. Daraus folgt, dass die Registrierung in der Slave-Phase der  $NCAP_C$ -STM Netzwerke abgeschlossen sein muss, bevor die Slave-Phase des  $NCAP_I$ - $NCAP_C$ -Netzwerkes endet.

Ab diesem Zeitpunkt warten die Slaves auf entsprechend zugeteilte Nachrichten, für die eine Antwort erwartet wird. Weiterhin ist die *MST-MTS*-Protokollebene so geschachtelt, dass der Kommunikationsvorgang innerhalb des  $NCAP_C$ - $STM_S$  Netzwerkes während der Kommunikation zwischen  $NCAP_I$  und  $NCAP_C$  abgeschlossen wird.

Somit wird die entsprechende Antwort zur Nachricht des  $NCAP_I$  mit den entsprechenden Informationen der  $STM_A$  oder  $STM_S$  an diesen Knoten übermittelt und die notwendigen Daten an die externe Strömungsregelung weitergegeben. Die als „TOKEN“ markierte Nachricht ist für die Kommunikation mit einem anderen  $NCAP_C$  bestimmt, der in diesem Plot nicht aufgetragen ist.

## Synchronisation

Es ist notwendig, die einzelnen Netzwerkteilnehmer zeitlich zu synchronisieren. Die Kommunikation kann durch zu hohe zeitliche Abweichungen gestört werden. Es ist möglich, dass die Kommunikationsregeln für die echtzeitige Kommunikation oder die zeitlichen Beschränkungen dadurch verletzt werden.

Für die Synchronisation der Knoten wird ein Zeitstempel in die Beacon-Nachricht implementiert. Vergleichbar mit der in [29] vorgeschlagenen Strategie bei der Messung der Paketübertragung, wird die Synchronisation per Broadcast umgesetzt. Für alle Ziel-Knoten wird dazu eine einheitliche Übertragungszeit angenommen.

Die Synchronisation wird vom Master-Knoten in jedem Kommunikationszyklus initiiert, seine Uhr wird als Referenz gesetzt. Dabei wird die aktuelle

Zeit des Master-Knotens den Slave-Knoten in den Beacon-Nachrichten mitgeteilt. So verfahren sowohl die  $\text{NCAP}_I$  mit den  $\text{NCAP}_C$ -Knoten als auch die  $\text{NCAP}_C$ -Knoten mit ihren STM-Slaves. Die Slave-Knoten passen ihre Uhr an, indem sie die Zeit übernehmen. Offensichtlich kommt es bei diesem Vorgehen zu einer Abweichung zwischen Master- und Slave-Knoten in der Größenordnung der Paketübertragungszeit. Das entspricht für die 64 Bytes großen Beacon-Pakete  $9,25 \mu\text{s}$ . Für den Ausgleich dieser Abweichung sind zusätzliche Paketübermittlungen, wie beispielsweise beim IEEE 1588 *Precision Time Protocol* (PTP), notwendig. Der IEEE Standard zu diesem Protokoll liefert eine detaillierte Beschreibung [39]. Im PTP werden die Übertragungszeit und die zusätzliche Abweichung der einzelnen Uhren voneinander durch den Austausch von mehreren Nachrichten ausgeglichen. Im vorliegenden Modell wird auf diesen zusätzlichen Aufwand verzichtet, da durch die identische Übertragungszeit alle Slave-Knoten nach Übernahme der vorgegebenen Zeit synchronisiert sind und die Genauigkeit der Synchronisation ausreicht. Die exakte Synchronisation mit dem Master-Knoten, also der Ausgleich der Übertragungszeit durch zusätzliche Nachrichtenübermittlungen, ist nicht notwendig.

## Fehlerbehandlung

Es ist vorgesehen, dass Fehler in der Slave-Phase dem Master mitgeteilt werden, sofern der fehlerhafte Knoten noch kommunizieren kann. Die Übertragung der Fehlermeldung erfolgt ähnlich der Registrierung beim Master. Die Fehler werden gegebenenfalls an die externe Regelung weitergegeben und je nach Anwendung müssen diese entsprechend behandelt werden.

Paketverluste in der Kommunikation werden durch Mechanismen im Transportprotokoll ausgeglichen. Dabei finden für den jeweiligen Paket-Typ eigene Lösungsansätze im Modell Anwendung.

Für den Verlust eines einzelnen Beacons muss die Kommunikation nicht vollständig zurückgesetzt werden, da diese ausschließlich zur Indikation von Beginn und Ende der Slave-Phase für die freie Kommunikation der Slaves verwendet werden. Wenn eine Nachricht vom Master an einen Slave verloren geht, so wird beim nachfolgenden Token eine Aufforderung zum erneuten Senden statt der Antwort des Slaves gesendet. Falls ein Acknowledgement in der ersten Phase der Kommunikation verloren geht, so wird die entsprechende Statusbenachrichtigung des Slaves in der nächsten Phase erneut gesendet. Es gibt zwei Fehlerfälle. Geht ein Token verloren, so wird es im nächsten Zyklus gesendet. Falls das Paket sich noch in der Abarbeitung befindet, so sendet der

Slave keine Antwort und der Master verlangt in der nächsten Token-Phase die Rückmeldung. Geht die Rückmeldung durch den Slave-Knoten verloren, so wird die Aufforderung zum erneuten Senden bei der nächsten Anfrage des Masters vom Slave zurückgesendet. Verluste in der Slave-Phase erhalten keine Bestätigung per Acknowledgement und werden dementsprechend in der nächsten Slave-Phase erneut gesendet. Besonders bei der Anmeldephase führt dies zu Verzögerungen, da die Nachrichten zur Anmeldung erneut gesendet werden müssen, um alle STM bei ihrem entsprechenden  $\text{NCAP}_C$  und alle  $\text{NCAP}_C$  bei ihrem zugehörigen  $\text{NCAP}_I$  zu registrieren.

### 3.4.4 Modellierung der Anwendungsschicht

Die Zusammenfassung der Applikation und Anwendungsschicht des drei Ebenen Modells ermöglicht die Implementierung der kompletten Anwendung innerhalb einer TrueTime-Code-Funktion. Bei der Implementierung auf einem Mikrocontroller ist die Trennung von Applikation und Anwendungsschicht vorgesehen.

Die Realisierung des Kommunikationsparadigmas erfolgt auf Basis des IEEE 1451.1 als Substandard des IEEE 1451. Die Vorgänge und Einzelheiten der IEEE 1451 Protokollfamilie sind für die Simulation in MATLAB/Simulink mit TrueTime nur schwer umsetzbar, aber die vollständige Implementierung ist auch nicht notwendig. Der Protokoll-Standard IEEE 1451.1 ist durch die Aufteilung in die drei Modellbereiche, Objektmodell, Datenmodell und Kommunikationsmodell, formal darstellbar. Details des IEEE 1451.1 sind in [86] beschrieben. Mithilfe vereinfachter Annahmen lässt sich der Standard mit den folgenden drei Aufgabenteilen modellieren:

1. Speicherung der knotenspezifischen Informationen mit einem elektronischen Datenblatt (TEDS)
2. Definition geeigneter Datentypen und eines geeigneten Paketrahmens zur Übertragung der notwendigen Daten
3. Festlegung der Übertragungszeit durch die IEEE 1451.1 Schicht

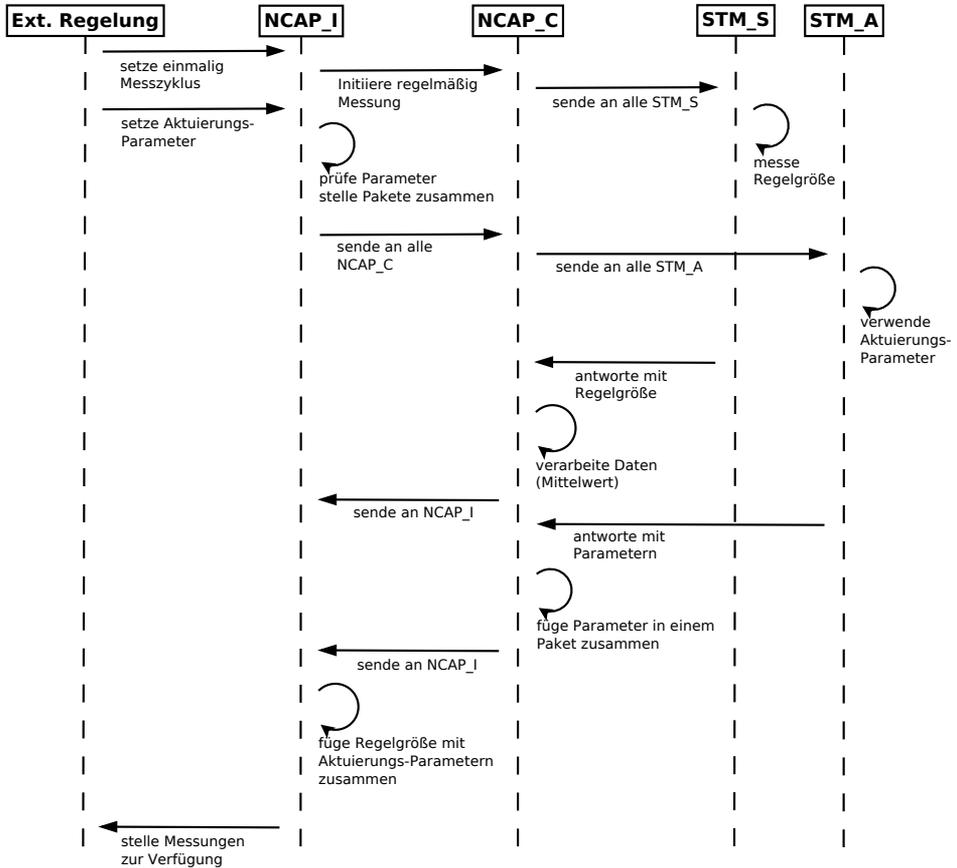
Die Informationen im elektronischen Datenblatt beinhalten den Knotentypen und den Abstand der Aktuator-Balken für die Berechnung der Phasendifferenz aus der Wellenlänge. Obere und untere Schranken für die Aktuierungsparameter sind ebenfalls hinterlegt. Darstellung und Inhalt stimmen nicht vollständig mit denen im Standard überein, die Informationen sind jedoch

problemlos erweiterbar. Die Parameter im Datenmodell werden durch primitive Datentypen repräsentiert und den Vorgaben aus TrueTime entsprechend in einer Struktur zusammengefasst. Alle hier verwendeten Datentypen sind mit denen im IEEE 1451.1 Protokoll vergleichbar. Die zwei verschiedenen Strategien zur Datenkommunikation (*Publish-Subscribe* und *Client-Server*) werden über Broadcast und Multicast-Nachrichten (Einer an alle oder einer an viele) sowie über eine direkte Adressierung abgebildet. Dabei werden die Knoten durch zugewiesenen Adressen oder über Broadcast- oder Multicast-Adressen angesprochen.

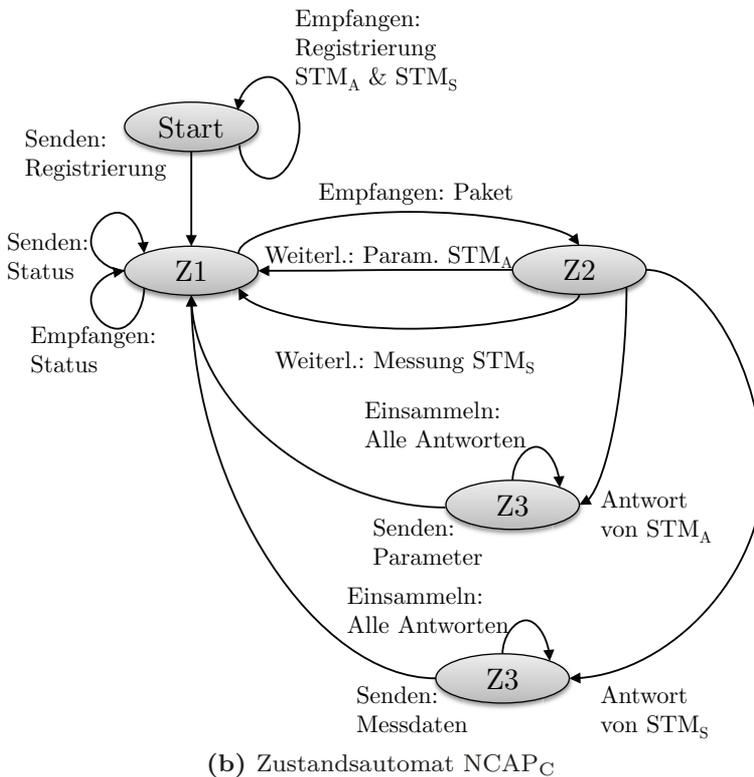
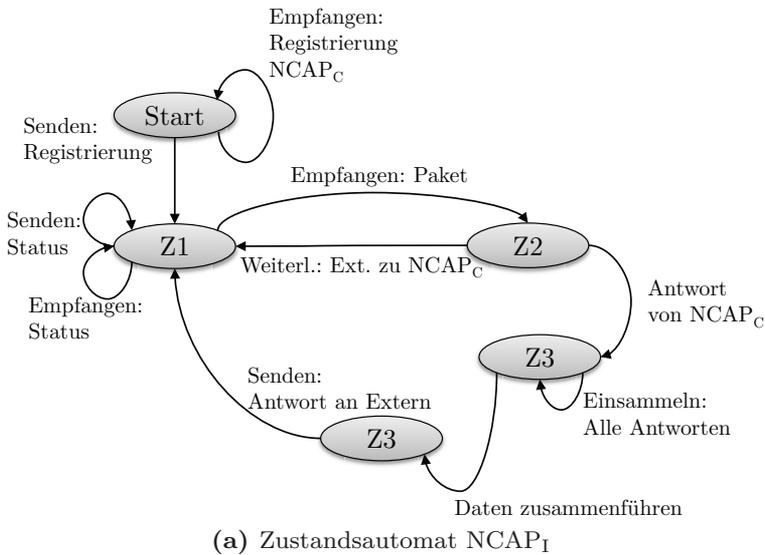
Zur Beschreibung der Anwendung werden Zustandsautomaten verwendet (siehe Abb. 3.11 und Abb. 3.12). Diese sind in den Kernel-Blöcken mithilfe der TrueTime-Code-Funktionen implementiert und die benötigte Ausführungszeit für die einzelnen Prozesse berücksichtigt. Jede Nachricht wird auf die Größe eines einzelnen Pakets im Rahmen der Ethernet-MTU von 1500 Bytes beschränkt. Damit wird der Overhead reduziert und die Paketsegmentierung muss nicht betrachtet und implementiert werden. Durch die Wahl der Baumstruktur entfällt ebenfalls die Implementierung eines Routing-Protokolls, da die Wege der Pakete klar definiert sind.

Es werden nicht die Rohdaten gesendet, sondern die Daten werden in einem entsprechenden Rahmen verpackt. Aus diesem lassen sich die notwendigen Informationen extrahieren. Die Identifikation erfolgt wie auf der Transportschicht über eine ID, diese ist identisch für Anfragen und zugehörige Antworten. Der Paketrahmen zur Übertragung der Daten enthält die Herkunfts- und die Zieladresse sowie einen Parameter zur Identifikation der Funktionszugehörigkeit. Abhängig von diesem Parameter, der die Werte **REGISTER**, **ACTUATE**, **MEASURE** und **STATUS** annehmen kann, werden die benötigten Parameter zur Registrierung, Aktuierung, Messung oder für Status- beziehungsweise Fehlermeldungen verwendet. Zur Überprüfung des Status der Abarbeitung werden die Nachrichtenpakete mit einem zusätzlichen Parameter (**valid**) indiziert. Neben der geschätzten Ausführungszeit der einzelnen Operationen in der Anwendung wird für die Übergabe eines Pakets an die Transportschicht eine Wartezeit von drei Millisekunden angenommen (siehe Kapitel 5.2.3). Der Datenaustausch mit der Transportschicht erfolgt dabei mit den zugehörigen Schnittstellenfunktionen für Master und Slave.

Zu Beginn der Simulation melden sich die einzelnen Knoten bei ihrem Master im Registrierungsprozess an. Die Slaves senden jeweils einen Broadcast auf das Netz und teilen so ihre Aufgabe ( $STM_A$ ,  $STM_S$ ,  $NCAP_C$ ) sowie ihre Adresse mit (siehe Abb. 3.10). Diese Daten werden von ihrem Master für die spätere Verwendung gespeichert. Die Anzahl der Teilnehmer ist in einem Modell fest



**Abb. 3.10:** Das Sequenzdiagramm zeigt den Verlauf der Kommunikation in der Anwendung innerhalb eines Kommunikationszyklus. Die externe Regelung überträgt die Messzykluszeit einmalig zu Beginn der Kommunikation und bei Bedarf die Aktuierungs-Parameter. Nach der Umsetzung der Parameter auf das Aktuator-System werden die gemessenen Parameter und die Messungen der Regelgröße zur Verfügung gestellt.

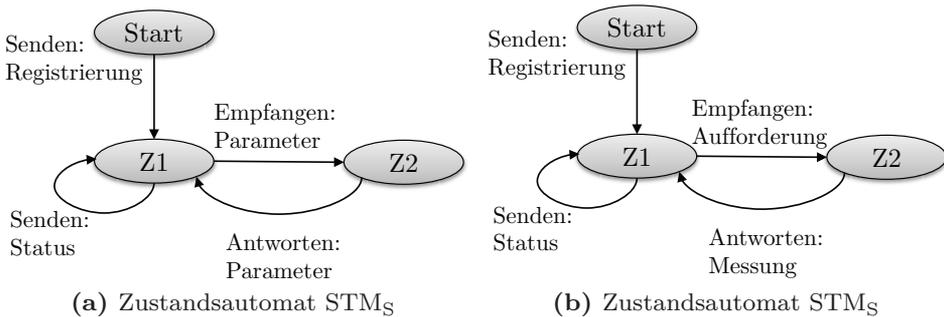


**Abb. 3.11:** Die Zustandsautomaten definieren die Verhaltensweisen der Knoten bei eintretenden Ereignissen für die NCAP-Knoten.

vorgegeben, sodass die Anwendung erst startet, wenn alle Teilnehmer sich bei ihrem Master registriert haben.

Die Abb. 3.10 zeigt den ersten Kommunikationszyklus nach der Registrierung der Knoten. Zuerst erwartet der  $\text{NCAP}_I$  ein Paket von der externen Regelung, das die Messzykluszeit beinhaltet. Die Messung wird dann entsprechend dieser Zykluszeit regelmäßig initiiert. Die zur Verfügung gestellten Stellgrößen der externen Regelung werden gegenüber vorgegebenen oberen und unteren Schranken geprüft. Sind die Daten korrekt, so werden sie in einer Nachricht vom  $\text{NCAP}_I$  an die zuständigen  $\text{NCAP}_C$  weitergeleitet. Ist der Bereichsparameter bei der Übergabe der Stellgrößen der externen Regelung nicht gesetzt, so werden alle  $\text{STM}_A$  mit dem gegebenen Parametersatz versorgt und es erfolgt eine einheitliche Aktuierung.

Die Nachricht wird in den  $\text{NCAP}_C$  empfangen und entsprechend der Sen-  
soraufforderung oder der Stellgrößen an die  $\text{STM}_S$  oder  $\text{STM}_A$  gesendet. Die Messung wird in den  $\text{STM}_S$ -Knoten durchgeführt beziehungsweise die Parameter werden im  $\text{STM}_A$ -Knoten umgesetzt und die tatsächliche Aktuierung gemessen.



**Abb. 3.12:** Die Zustandsautomaten definieren die Verhaltensweisen der Knoten bei eintretenden Ereignissen für die  $\text{STM}$ -Knoten.

Die erfassten Daten zum Reibungswiderstand und zur Aktuierung werden jeweils zurück zum  $\text{NCAP}_C$ -Knoten gesendet, wo sie für jede Gruppe  $\text{STM}_A$  und  $\text{STM}_S$  zusammengefasst und an den  $\text{NCAP}_I$ -Knoten weitergeleitet werden. Dieser erwartet eine Antwort von jedem der Netzwerkteilnehmer  $\text{NCAP}_C$  und stellt dann für jeden Bereich je ein Datenpaket der externen Strömungsregelung zur Verfügung. Sind die Daten beim Eingang nicht vollständig, werden entweder standardisierte Parameter angenommen, wie beispielsweise für den

Bereichsparameter, oder es wird eine Fehlermeldung an die Strömungsregelung zurückgegeben.

Die einzelnen Komponenten müssen aufeinander abgestimmt werden, damit die Kommunikation innerhalb eines *MTS*-Zyklus abgeschlossen ist. Die Randbedingungen der Aktuierung müssen beachtet werden, beispielsweise ist es möglich, ein Einschwingverhalten mithilfe des *MTS*-Protokolls in einer verlängerten Master-Phase zu berücksichtigen.

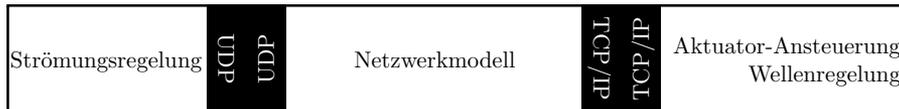
## Fehlerbehandlung

Die Stellgrößen der externen Regelung müssen auf Plausibilität geprüft werden. Liegen die Daten nicht innerhalb der durch den inneren Regelkreis vorgegebenen Grenzen, so werden vordefinierte Fehlercodes über den Fehlerparameter an die externe Regelung übergeben. Diese ist dann für die Korrektur des Fehlers zuständig.

Ausfälle müssen für den jeweiligen Knotentypen einzeln beschrieben werden. Ein Fehler im  $STM_A$  hat im schlimmsten Fall den Ausfall eines kompletten Bereichs zur Folge, wenn ein Aktuator durch einen mechanischen Defekt nicht mehr seiner Aufgabe nachkommen kann und einen vollständigen Bereich mit seiner Fehlfunktion beeinflusst. Ist der Aktuator nicht mehr ansteuerbar, aber weiterhin in gewisser Weise funktionsfähig, so kann die Aktuierung gegebenenfalls unter Anpassung der Parameter fortgesetzt werden. Ein Hardware-Fehler der analogen Ansteuerung kann nur durch eine zusätzliche Redundanz ausgeglichen werden. Ein Ausfall eines  $STM_S$  beziehungsweise fehlerbehaftete Messungen lassen sich gegebenenfalls durch eine geschickte Signalverarbeitung ausgleichen. Die Aufgaben eines ausgefallenen  $NCAP_C$  kann von einem anderen Knoten übernommen oder auf mehrere Knoten verteilt werden. Ein Ausfall eines Schnittstellenknotens  $NCAP_I$  zur Strömungsregelung kann entweder durch einen Reserveknoten oder ebenfalls durch ein vollvermaschtes Netz aufgefangen werden, wenn dann einer der  $NCAP_C$  die Aufgaben des  $NCAP_I$  zusätzlich übernimmt. Dies erfordert jedoch zusätzliche Rechenleistung, die nicht zwingend zur Verfügung steht.

### 3.4.5 Model-in-the-Loop-Implementierung

Die Realisierung der Strömungsregelung im IRT erfolgt auf einem Echtzeitsystem der Firma Speedgoat [80] mit MATLAB und Simulink. Die Aktuator-



**Abb. 3.13:** Die definierten Schnittstellen ermöglichen den einzelnen Komponenten eine standardisierte Kommunikation ohne die genaue Funktionsweise der anderen Komponenten kennen zu müssen. Für die Kommunikation werden etablierte Netzwerkprotokolle eingesetzt.

Die physikalische Verbindung erfolgt über ein Ethernet. Es ist sowohl ein switched Ethernet als auch eine direkte Verbindung über ein Crossover-Kabel möglich. Für die Parameterstudien im Windkanal wird dazu direkt die von National Instruments bereitgestellte Benutzerschnittstelle verwendet. Für die Model-in-the-Loop-Einbindung des Netzwerkes werden Netzwerkschnittstellen entsprechend Abb. 3.13 definiert.

Das Netzwerkmodell ist auf der einen Seite per UDP an die externe Regelung angebunden und auf der anderen Seite per TCP/IP an die Aktuator-Ansteuerung. Zwischen der externen Regelung und dem Netzwerkmodell wird eine Punkt-zu-Punkt-Kommunikation etabliert, sodass es sich anbietet, ein verbindungsloses Protokoll zu verwenden. Im Gegensatz zur externen Regelung als einzelne Komponente ist vorgesehen, mehrere Aktuator-Systeme in einem Netzwerk für die Model-in-the-Loop-Verwendung anzubinden oder miteinander kommunizieren zu lassen, sodass die Verwendung des TCP/IP-Protokolls nahe liegt. Der Einsatz etablierter Protokolle stellt sicher, dass die softwareseitigen Bibliotheken zur Netzwerkkommunikation vorhanden sind und eine schnelle Verfügbarkeit der Schnittstelle garantiert ist.

**Aufistung 3.2:** Aufbau eines beispielhaften XML-Schemas.

```
<ID parameter -1="value-1" parameter -2="value-2" ...
/>
```

Die Nachrichtendarstellung innerhalb der Anwendungsschicht wird zwischen der externen Regelung und der Netzwerksimulation in einem flexiblen XML-basierten Datenformat abgewickelt. Eine detaillierte Beschreibung von XML ist in [47] beschrieben. Es werden ASCII-basierte Nachrichten im vorgegebenen XML-Schema ausgetauscht (siehe Auflistung 3.2), wobei die Reihenfolge der Parameter in diesem Format nicht festgelegt ist. Die angesprochene Identifikation der Nachrichten wird mittels Identifikationsnummer (ID) durchgeführt, die dem XML-Tag vorangestellt wird. Das XML-Schema ermöglicht die flexible Erweiterung der möglichen Parameter, sodass die allgemeinen Parameter zur Kommunikation und die speziellen Parameter, definiert durch die externe Regelungsaufgabe, jederzeit anpassbar sind (siehe Auflistung 3.3).

**Auflistung 3.3:** Beispiel des verwendeten XML-Schemas.

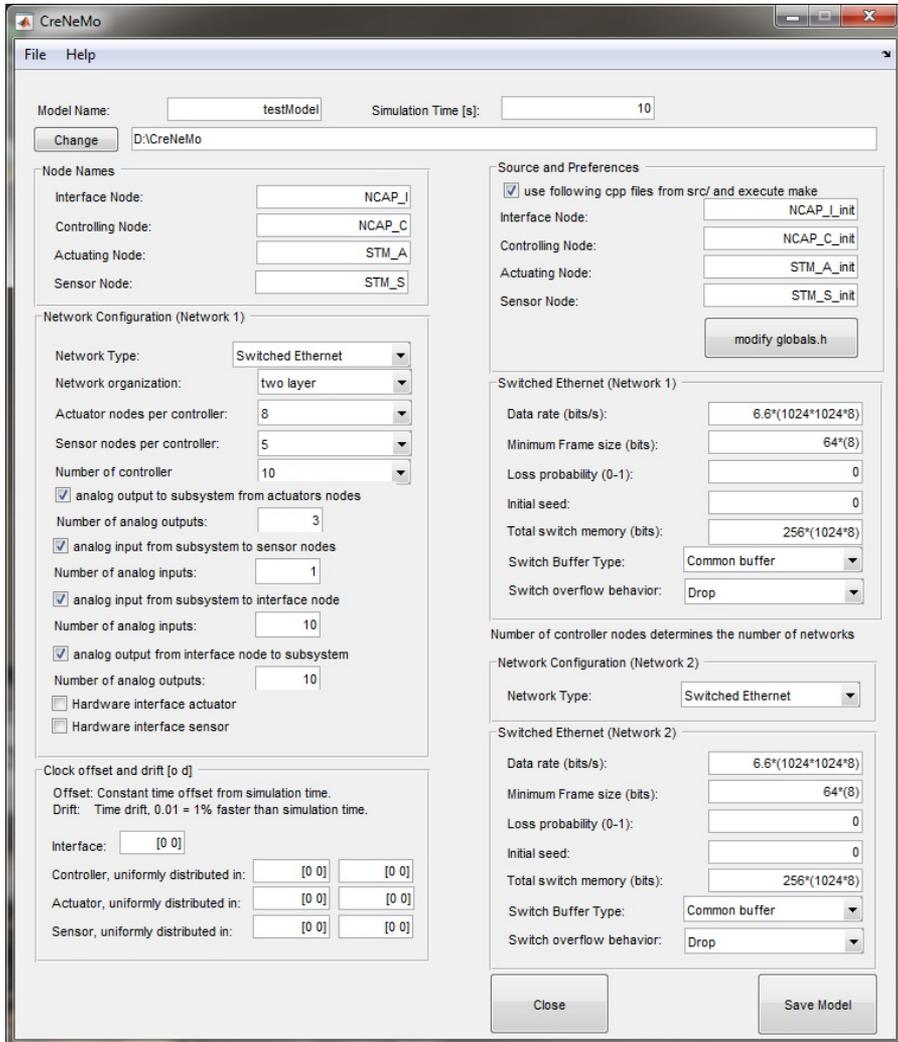
```
1 <1001 area="2" accucontrol="0,05"  
   sampletimesens="250" />
```

Für die Ansteuerung der Aktuatorik ist die Generierung von Wellensignalen in Abhängigkeit von gegebenen Wellenparametern implementiert und in einem einzelnen DAQ-System zur Verwendung im Windkanal modelliert. Durch die Wahl des TCP/IP Protokolls werden Probleme bei der mechanischen Aktuierung ausgeschlossen, die auf Fehler aus der Datenübertragung zurückzuführen sind. Die vorgesehene Schnittstelle dient dabei dem Informationsaustausch von Aktuierungs-Parametern, Messwerten und möglichen Fehlern. Daher wird die Kommunikation auf die notwendigen Parameter in ASCII-Form zwischen dem Netzwerkmodell und der Aktuator-Ansteuerung reduziert. Weitere gegebenenfalls notwendige Parameter lassen sich mit geringem Aufwand in die Kommunikationsschnittstellen integrieren.

## 3.5 Modellkonfiguration und Simulation

Die Implementierung der Netzwerkmodelle wurde auf einem Lenovo T430 mit einem Intel i7-3520M (2,9 GHz) und 16 GB RAM durchgeführt. Die MATLAB-Umgebung in der Version 2015b (32-bit) wurde zusammen mit der Simulink-Toolbox auf dem Betriebssystem Windows 7 x64 benutzt. Das Modell wurde mithilfe der Toolbox TrueTime 2.0 beta 7 erstellt und simuliert.

Die Simulationsauswertung erfolgt über die Analyse der Ausgabe-Signale der Netzwerk-Blöcke und der Kernel-Blöcke sowie dem Vergleich von eingehenden



**Abb. 3.14:** Die grafische Benutzeroberfläche ermöglicht die Konfiguration und Generierung von Netzwerkmodellen mithilfe der TrueTime-Toolbox in MATLAB/Simulink. Sämtliche Parameter sowie Quellcode und Subsysteme können integriert und kompiliert werden, sodass die Simulation sofort einsatzbereit ist.

und ausgehenden Daten aus der Sicht des  $\text{NCAP}_1$ . Der Verlauf der Simulationen wird mithilfe der Scopes visualisiert (siehe Kapitel 2.4). Innerhalb der Simulationen werden Kommunikationsfehler und noch nicht verarbeitete Pakete im Eingangspuffer der Knoten am Simulationende gezählt. Durch eine geschickte Wahl der Simulationslänge zum Ende eines Zyklus oder einer Kommunikationsphase bezogen auf das *MTS*-Protokoll lässt sich die Bearbeitung der Pakete beurteilen.

In der Regel beinhalten die Simulationen 50 *MTS*-Zyklen, um die Funktionsweise des Netzwerkes zu evaluieren. Die Modellparameter und Verhaltensweisen werden als fest für eine Simulation definiert.

Da die Erstellung großer Netzwerke mithilfe von MATLAB/Simulink und TrueTime manuell sehr aufwändig ist, wurde im Rahmen dieser Arbeit ein Werkzeug zur Generierung dieser Netzwerke implementiert. Es trägt den Namen *CreNeMo* (Create Network Model) und erzeugt einfach und schnell Simulink-Modelle mit der gewünschten Anzahl an Knoten mithilfe der TrueTime-Toolbox.

Das Tool ermöglicht die Festlegung verschiedener Parameter für das Modell bereits vor der Modellgenerierung (siehe Abb. 3.14). Als Parameter sind sowohl die Namen als auch die Anzahl der Knoten festgelegt. Es kann ein Netzwerktyp (MAC-Protokoll) und eine Organisationsstruktur ausgewählt werden (eine Ebene, zwei Ebenen, vollvermascht). Die zeitliche Abweichung der Knoten gegenüber der Simulationszeit ist veränderbar. Es gibt die Option, Schnittstellen zu Simulink-Subsystemen zu definieren. Diese lassen sich direkt bei der Erzeugung der Simulation in das Modell einbinden. Bereits vorgefertigte Quelldateien, die das Verhalten der einzelnen Knoten in der Simulation beschreiben, werden bei der Modellgenerierung eingebunden und kompiliert. Die Modifikation verschiedener Parameter für das beschriebene Modell wird in der Datei `globals.h` vorgenommen. Die Parameter für die Netzwerke in der Oberfläche entsprechen denen in den Netzwerkknoten der TrueTime-Toolbox.

## 3.6 Zusammenfassung

Das entwickelte Netzwerkkonzept berücksichtigt sowohl die allgemeinen Anforderungen für die großflächige Umsetzung von Regelungsaufgaben mithilfe von Aktuator- und Sensornetzwerken als auch die Applikationsanforderungen der Strömungsregelung mittels transversaler Oberflächenwellen auf großen

Transportmitteln im hohen Reynoldszahlenbereich. Dabei wird die Aufteilung der Aufgaben in Kontroll-, Aktuator- und Sensor-Knoten definiert und das Netzwerk sowie die Ansteuerung der Aktuatorik als Teil des Projektes in einen kaskadierten Regelkreis eingeordnet.

Gemäß OSI-Referenzmodell wird die Netzwerkkommunikation mit einem Protokollstapel in drei Schichten beschrieben. Dieser besteht aus Anwendungsschicht, Transportschicht und Netzwerkschicht. Die Netzwerkschicht definiert die Übertragung über ein switched Ethernet Netzwerk in einer Baumstruktur. Innerhalb der Transportschicht kommt das selbstentwickelte *Master-Token-Slave*-Protokoll zum Einsatz. Darin wird der Kommunikationszyklus in drei Phasen aufgeteilt und es ist eine deterministische Kommunikationsstrategie implementiert. Die Anwendungsschicht ist in Anlehnung an den IEEE 1451 Standard realisiert und eine ereignisbasierte Applikation ist in jedem Knoten implementiert.

Ein Modell zur Validierung des Konzeptes ist in MATLAB/Simulink mithilfe der TrueTime-Toolbox implementiert. Zur flexiblen, parametrisierten Generierung großflächiger Aktuator- und Sensornetzwerkmodelle wurde das Werkzeug *CreNeMo* entwickelt. Mithilfe gängiger von Simulink bereitgestellter Methoden können die Netzwerksimulationen analysiert und unterschiedliche Netzwerkkonfigurationen verglichen werden.



# 4 Erzeugung von Oberflächenwellen im aktuatornahen Regelkreis

In diesem Kapitel wird die Erzeugung von geregelten Oberflächenwellen mithilfe des elektromagnetischen Aktuator-Systems beschrieben. In enger Zusammenarbeit mit den Projektpartnern der Forschergruppe wurde sowohl die Hardware als auch die Software für die Durchführung der Parameterstudien im Windkanal durch den Projektpartner AIA und für die Validierung der externen Strömungsregelung des Projektpartners IRT realisiert. Zur Sicherstellung der Umsetzung der Wellenbewegung wird ein Regelungskonzept zur Regelung der einzelnen Aktuatoren mit einer Genauigkeit von fünf Prozent bezogen auf das vorgegebene Ansteuerungssignal vorgestellt.

Im Folgenden wird das elektromagnetische Aktuator-System zur Erzeugung der transversalen Oberflächenwellen, welches im Rahmen des Projektes im ZEA-2 entwickelt wurde, in Anlehnung an die Beschreibung in Schiek et al. [73] vorgestellt. Die weiteren Inhalte dieses Kapitels finden sich zu unterschiedlichen Teilen in diesen Veröffentlichungen von Dück et al.: [31], [25].

## 4.1 Elektromagnetisches Aktuator-System zur Erzeugung von Oberflächenwellen

Gemäß den Projektanforderungen wurde ein Aktuator-System zur Erzeugung von sinusförmigen Oberflächenwellen mit einer Frequenz von bis zu 200 Hz entwickelt (siehe Abb. 4.1). Das Aktuator-System basiert auf Luftspulen mit 200 Windungen Kupferdraht (Durchmesser 0,12 mm), die auf einen PCB-Träger verklebt jeweils in einem statischen Magnetfeld erzeugt von starken Permanentmagneten gelagert sind (siehe Abb. 4.2). Die Permanentmagnete sind in zehn Millimeter starken Aluminiumträgern fixiert und erzeugen ein

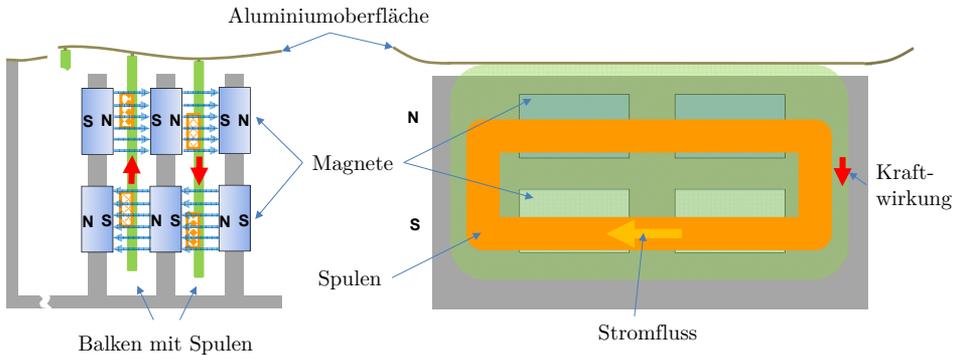
Magnetfeld von etwa 0,8 T im Bereich der Spulen. Der Abstand zwischen den einzelnen Luftspulen beträgt 20 mm.



**Abb. 4.1:** Seitenansicht des PCB-Luftspulen-Aktuator-Systems ohne Aluminiumoberfläche.

Der Durchfluss eines positiven oder negativen Stroms resultiert in den horizontal gelagerten Balken nach dem Prinzip der Lorenzkraft in einer vertikalen Kraft. Die erzeugte Kraft wird auf die angeklebte Aluminiumoberfläche von bis zu 0,5 mm Stärke weitergegeben. Durch die Ansteuerung mit periodischen phasenverschobenen Signalen lässt sich eine laufende, transversale Oberflächenwelle erzeugen [31]. Dabei fungieren die einzelnen Balken als Stützstellen der Welle. Die Auslenkung der Balken wird über Lichtschrankensensoren an der Unterseite mit einer Genauigkeit besser als  $0,1 \mu\text{m}$  erfasst.

Beim gegebenen Aktuator-Abstand von 20 mm kann im Prinzip für eine wandernde Welle eine minimale Wellenlänge von 80 mm erzeugt werden. Um eine glatte Sinusform dieser wandernden Welle sicherzustellen wird jedoch mit einer minimalen Wellenlänge von 160 mm gearbeitet, dies entspricht einer Phasenverschiebung zwischen den Aktuatoren von 45 Grad. Bei dieser Wellenlänge würde der Bereich der elastischen Verformung der Aluminiumoberfläche erst ab einer Wellenamplitude von 2,4 mm überschritten werden. Das System wird mit einer Kombination aus einem leistungsstarken HiFi-Verstärker (Thomann, TSA 4-1300 [83]) für die Wellenerzeugung und einer selbstentwickelten DC-Stromquelle für die Justierung der Mittellage der Oberfläche betrieben. Dabei wird die Oberflächenwelle mithilfe periodischer, phasenverschobener Signale verschiedener Amplituden und Frequenzen gesteuert. Die bisherigen Messungen im subsonischen Windkanal in [72] und [52] wurden mithilfe des



**Abb. 4.2:** Übersicht der Kupferspule auf einem PCB-Balken im Aktuator-System. Durch Anlegen einer positiven oder negativen Spannung an der Spule wird eine Kraft in vertikaler Richtung erzeugt, die den Balken und damit die angeklebte Aluminiumoberfläche bewegt [73].

Systems durchgeführt.

$$F_{\max} = -\frac{E \cdot Q}{d} \cdot 2\Delta x \left( 1 - \frac{1}{\sqrt{1 + \left(\frac{\Delta x}{d}\right)^2}} \right) \quad (4.1)$$

Für die Auslenkung der Oberfläche lässt sich die notwendige Kraft nach Gleichung 4.1 bestimmen. Dabei fließen die Elastizitätskonstante  $E$  des Aluminiums ( $70 \frac{\text{kN}}{\text{mm}^2}$ ) und die Querschnittsfläche  $Q$  der gedehnten Aluminiumoberfläche ( $300 \text{ mm} \cdot 0,3 \text{ mm}$ ) ein. Der Balkenabstand  $d$  beträgt  $20 \text{ mm}$ . Die Größe  $\Delta x$  beschreibt die Differenz der Auslenkung zwischen den benachbarten Aktuatoren. Für laufende sinusförmige Oberflächenwellen mit einer Länge von  $160 \text{ mm}$  und der Amplitude von einem Millimeter beträgt diese Differenz maximal  $0,3 \text{ mm}$ . Entsprechend der Gleichung 4.1 wird dazu eine maximale Kraft von etwa  $20 \text{ N}$  benötigt.

$$F = I \cdot l \cdot B \quad (4.2)$$

Die Lorentzkraft  $F$  auf einen stromdurchflossenen Leiter der Länge  $l$  berechnet sich bei einem Strom  $I$  und senkrecht zum Leiter wirkenden Magnetfeld mit der Flussdichte  $B$  gemäß der Gleichung 4.2. Wie der Abbildung 4.1 zu

entnehmen ist, überdeckt die auf den PCB-Träger aufgeklebte Spule insgesamt acht Permanentmagnete. Bei einer Länge der Magnete von fünf Zentimetern ergibt sich eine Gesamtlänge des Leiters innerhalb des Magnetfeldes von 40 cm. Das magnetische Feld zwischen den Permanentmagneten wurde mit einem Fluxmeter mit 0,8 T gemessen. Die Spulen weisen mit ihren 200 Wicklungen einen ohmschen Widerstand von 160  $\Omega$  auf. Der verwendete Verstärker kann somit (im Frequenzbereich von 40 Hz bis über 15 kHz) einen maximalen Strom von etwa 0,35 A erzeugen. Damit ergibt sich theoretisch eine maximale Aktuierungskraft von 24 N pro Aktuator. Experimentell wurde die maximale Kraft mit 20 N bestimmt, die Differenz zum theoretischen Wert erklärt sich unter anderem aus der idealisierten Feldannahme in der Berechnung.

Ein neues System befindet sich derzeit in der Entwicklung. Durch Verkürzung des Balkenabstandes auf zehn Millimeter ist die Erzeugung von kürzeren Wellenlängen möglich. Außerdem konnte durch Verwendung von dickeren Spulendraht bei 80 Wicklungen der ohmsche Widerstand auf 15  $\Omega$  gesenkt werden. Dies passt deutlich besser zur Leistungscharakteristik des Verstärkers, der nunmehr bis zu fünf Ampere Strom durch die Spule leiten kann. Bei einer Beschränkung auf maximal drei Ampere kommt das gesamte System ohne aktive Kühlung aus und die maximale Aktuatorkraft kann trotzdem auf über 60 N verdreifacht werden. Die Kopplung des Verstärkerausgangs mit der selbstentwickelten DC-Stromquelle erlaubt zudem die Justierung der Mittel-lage der Oberfläche.

Für die Ansteuerung der Windkanalversuche wird das National Instruments PXI System PXIe-8133 mit den Einschubkarten PXI-6723 und PXI-6255 verwendet [61]. Das System verfügt aktuell über 64 individuell ansteuerbare analoge Ausgänge (13 Bit) und 80 analoge Eingänge (16 Bit) mit einer Summenabtastung von 750  $\frac{\text{KS}}{\text{s}}$ , ist jedoch modular deutlich erweiterbar.

## 4.2 Echtzeitige Parametervariation für die Oberflächenwellen (Open Loop)

Das oben beschriebene Aktuator-System definiert die Randbedingungen für die Implementierung einer echtzeitigen Variation der Oberflächenwellen. Dies betrifft die Bereitstellung der richtigen Ansteuerungssignale, die Sicherstellung der korrekten Aktuierungsform sowie auch die Verarbeitung der zeitsynchron erfassten Messdaten.

## 4.2 Echtzeitige Parametervariation für die Oberflächenwellen (Open Loop)

---

Als Grundlage zur Ansteuerung der Aktuatoren dient das Sinussignal aus Gleichung 4.3 mit den Wellenparametern Amplitude ( $A$ ), Frequenz ( $F$ ), Phasenverschiebung ( $P$ ) und Offset ( $O$ ). Die Phasenverschiebung ergibt sich nach Gleichung 3.2 aus der gewünschten Wellenlänge.

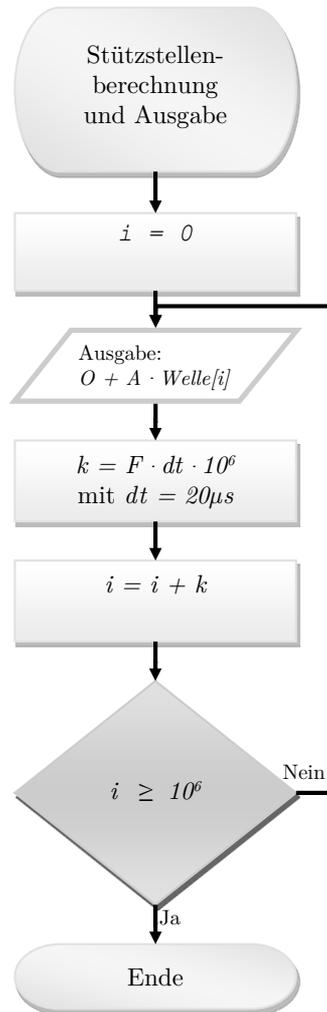
$$S(t) = A \cdot \sin(F \cdot 2\pi \cdot t + P) + O \quad (4.3)$$

Wie in [26] beschrieben, können gewünschte Parameteränderungen (Frequenz und Amplitude) auch ohne Verwendung eines speziellen Echtzeitsystems während der laufenden Aktuierung innerhalb von zwei Wellenperioden umgesetzt werden. Die Umsetzung wird durch das verwendete nicht echtzeitige Betriebssystem zeitlich nicht beeinflusst. Eine Phasenänderung wird über zwei aufeinanderfolgende Frequenzänderungen realisiert und benötigt daher insgesamt vier Perioden. Die benötigte Zeit für eine Parameteränderung ist damit von der aktuellen Aktuierungs-Frequenz abhängig.

Die Implementierung erfolgt unter Ausnutzung des Ringpuffer-Speichers [60]. Diese Strategie ermöglicht kontinuierliche Parameterstudien durch eine echtzeitige Änderung der Oberflächenwellen im Windkanal. Implementiert wurde diese Ansteuerung auf dem oben beschriebenen National Instruments DAQ-System.

Auf dem gleichen System wurde als Weiterentwicklung die Aktuator-Ansteuerung auf Basis des LabVIEW Real-Time Betriebssystems implementiert. Die Verwendung dieser Echtzeitumgebung in Kombination mit einer modifizierten Pufferstrategie ermöglicht die Verringerung der Übergangszeit zwischen zwei verschiedenen parametrisierten Wellen auf weniger als eine Periode. Darüber hinaus wird die Aufzeichnung sämtlicher Überwachungsdaten für das Aktuator-System vereinfacht und weiterhin Rechenkapazität für eine zusätzliche Regelungsaufgabe vorgehalten. Im Zusammenhang mit dem in Kapitel 3 vorgestellten Netzwerkkonzept und dem entsprechenden Netzwerkmodell repräsentiert diese LabVIEW Real-Time Implementierung eine Verknüpfung von  $STM_A$  und  $STM_S$ . Eine detaillierte Beschreibung dieser Entwicklung und der zugehörigen Software findet sich in Völkel et al. [87]. Das System ist in einzelne Module aufgeteilt:

1. Kommunikation
2. Dateisystem
3. Wellenansteuerung
4. Datenverarbeitung



**Abb. 4.3:** Generierung und Ausgabe einer einzelnen Periode des Ansteuerungs-Signals. Die Wertetabelle *Welle* wird verwendet um in Abhängigkeit von Amplitude (*A*), Offset (*O*) und Frequenz (*F*) das entsprechende Signal zu erzeugen.

## 4.2 Echtzeitige Parametervariation für die Oberflächenwellen (Open Loop)

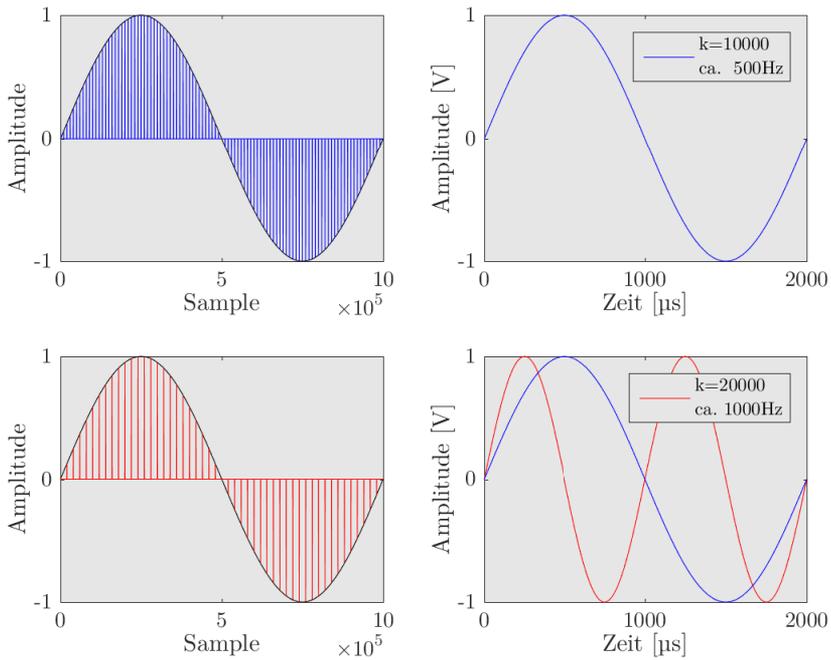
---

Der modulare Aufbau erleichtert die sukzessive Erweiterung der Aktuator-Ansteuerung. Das Modul *Kommunikation* stellt die Verbindung zur Benutzerschnittstelle dar für das Empfangen der Parameter zur Wellengenerierung. Im *Dateisystem* werden die nötigen Wertetabellen abgelegt. Die *Wellensteuerung* stellt den Auslesemechanismus sowie die automatisierte Erzeugung glatter Transienten zur Verfügung. Die Ausgabe der analogen Wellensignale und Aufnahme der Sensordaten sowohl für die Wellenregelung als auch für die externe Strömungsregelung wird in der *Datenverarbeitung* durchgeführt. Da die kontinuierliche Berechnung einzelner Stützstellen für die wellenförmige Aktuierung sehr rechenintensiv ist [87], werden die notwendigen Wellenformen (Sinus, Dreieck, usw.) beim Start der Software einmalig erzeugt und als Wertetabelle abgelegt. Die einzelnen Stützstellen werden nach dem Auslesen mit der vorgegebenen Amplitude multipliziert und entsprechend des vorgegebenen Offsets korrigiert und anschließend an den DAC zur Ausgabe weitergegeben. Eine solche hochauflösende Repräsentation einer einzelnen Periode einer bestimmten Wellenform ermöglicht durch flexible Abtastung die Ausgabe verschiedener Wellen unterschiedlicher Frequenzen.

### 4.2.1 Generierung glatter Signalübergänge

Die Anpassung der gespeicherten Periode der Wellenform für das entsprechende auszugebende Signal erfolgt durch Additions- beziehungsweise Multiplikationsoperationen. Dies erspart die rechenaufwändige Auswertung trigonometrischer Funktionen und ermöglicht zudem auch die Verwendung beliebiger Wellenformen. Eine flexible Abtastung der gespeicherten Wellenperiode ermöglicht sowohl eine Frequenz- als auch eine Phasenänderung. Abb. 4.3 beschreibt die Ausgabe einer Signalperiode. Die Schrittweite  $k$  wird maßgeblich durch die Frequenz der auszugebenden Welle bestimmt. Der ausgelesene Abtastwert  $Welle[i]$  wird mit dem Wert der Amplitude multipliziert und der Offset dazu addiert.

Eine kurzzeitige flexible Änderung der Abtastung ermöglicht eine Phasenverschiebung. Dazu wird die Schrittweite  $k$  modifiziert. Für ein konstantes  $k$  wird eine Welle mit konstanter Frequenz erzeugt (siehe Abb. 4.3 und Abb. 4.4). Zur Erzeugung möglichst glatter Übergänge zwischen unterschiedlich parametrisierten Wellen werden die Parameteränderungen nur an dafür sinnvollen Signalpunkten durchgeführt. So werden Amplituden- und Frequenzänderungen beispielsweise erst im Nulldurchgang übernommen, da dort die Kraft im Aktuator-System verschwindet.

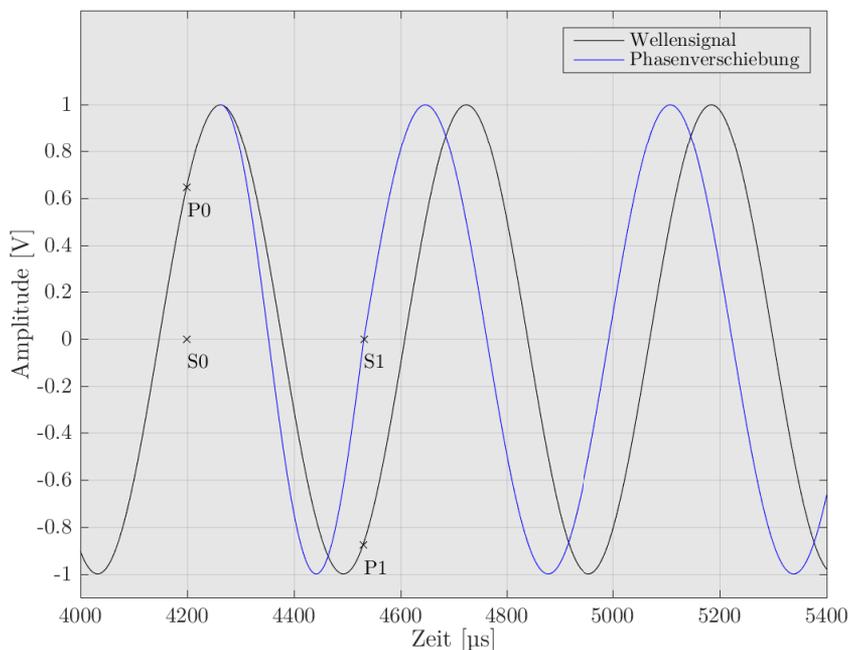


**Abb. 4.4:** Die Berechnung einer Stützstelle des Signalverlaufs erfolgt auf Basis des ausgelesenen Wertes aus der Wertetabelle und der Schrittweite  $k$ , wobei  $k$  durch die gegebene Frequenz und die Taktrate des DAQ-Systems bestimmt wird.

## 4.2 Echtzeitige Parametervariation für die Oberflächenwellen (Open Loop)

Wird  $k$  beim Auslesen der Wertetabelle für einige Zeitschritte erhöht und danach auf den Ausgangswert zurückgesetzt, so wird die Phase der Welle modifiziert (siehe Abb. 4.5). Die Berechnungen für die Signalübergänge an den einzelnen Punkten sind im Folgenden dargestellt:

Bis $(S_0, P_0)$ :	$k = F \cdot dt \cdot 10^6$	mit $dt = 20 \mu s$
Von $(S_0, P_0)$ bis $(S_1, P_1)$ :	$k = F \cdot dt \cdot 10^6 + \delta$	mit $dt = 20 \mu s, \delta > 0$
Ab $(S_1, P_1)$ :	$k = F \cdot dt \cdot 10^6$	mit $dt = 20 \mu s$



**Abb. 4.5:** Schematisches Beispiel zur unterbrechungsfreien Phasenverschiebung einer kontinuierlichen analogen Sinuswelle durch zwischenzeitliche Erhöhung der Frequenz.

Die Änderung der Wellenform ist nur jeweils zu Beginn einer Periode sinnvoll. Offsetänderungen in positive Richtung werden bei steigenden-, in negative Richtung nur bei fallenden Abschnitten der Wellenform übernommen. Diese

Strategien ermöglichen für die Änderungen von Wellenparametern eine garantierte Reaktionszeit von weniger als einer Periode. Für die Änderungen von Amplitude, Frequenz und Offset ist weniger als eine halbe Periode möglich.

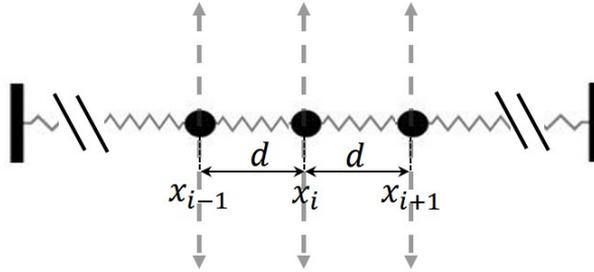
Die in diesem Kapitel vorgestellten Konzepte für die Aktuator-Ansteuerung inklusive der echtzeitigen Variation der Wellenparameter basieren auf der Annahme, dass die über die Aluminiumoberfläche übertragene Kopplung zwischen den Aktuatoren gegenüber der Reibung der Aktuator-Balken zu vernachlässigen ist und die einzelnen Aktuator-Balken daher im Rahmen der anvisierten Genauigkeit dem Ansteuerungssignal folgen. Für das bislang verwendete Aktuator-System konnte diese Annahme für die verwendeten Wellenparameter (Amplitude  $A \leq 0,5$  mm, Wellenlänge  $\lambda = 160$  mm, Frequenz  $F = 81$  Hz) experimentell bestätigt werden.

Es zeigt sich jedoch, dass bei größeren Amplituden und Frequenzen beziehungsweise kleineren Wellenlängen diese Annahme nicht mehr erfüllt ist. Da mit dem neuen Aktuator-System Oberflächenwellen in diesem Parameterbereich erzeugt werden sollen, muss das System eingehender betrachtet werden und der Einfluss der Kopplung durch die Aluminiumoberfläche auf die Aktuator-Bewegung und damit auf die erzeugten Oberflächenwellen zunächst im Modell untersucht werden. Die bisherigen Oberflächenwellen im Windkanal wurden mithilfe einer ausgiebigen Kalibrierung realisiert.

### 4.3 Aktuator-Modell

Vereinfacht lassen sich die Aktuatoren als eindimensionale harmonisch getriebene, gedämpfte Masse-Feder-Systeme beschreiben, wobei die elastische Rückstellkraft über die Ausdehnung der Aluminiumoberfläche gleichzeitig die Kopplung zwischen den Aktuatoren beschreibt. Der technische Aufbau des Aktuator-Systems ist in Kapitel 4.1 erläutert. Die Darstellung als eindimensionaler Oszillator in Abb. 4.6 folgt der Differentialgleichung 4.4 [31]. Dabei beschreibt  $m_i$  die Masse der Aktuatoren und der bewegten Aluminiumoberfläche und  $x$  die Regelgröße, die messbare vertikale Auslenkung der Aktuatoren.  $F_i$  ist die magnetische Kraft (Lorentzkraft und Gewichtskraft). Sie ist proportional zur Stellgröße, dem Stromfluss beziehungsweise der Spannung.  $F_i^r$  ist die Rückstellkraft, die durch die Kopplung über die Aluminiumoberfläche auf die bewegten Aktuatoren wirkt, und  $r_i$  beschreibt den Reibungskoeffizienten.

$$m_i \cdot \ddot{x}_i = F_i(t) - F_i^r(x_i, x_{i-1}, x_{i+1}) - r_i \cdot \dot{x}_i \quad (4.4)$$

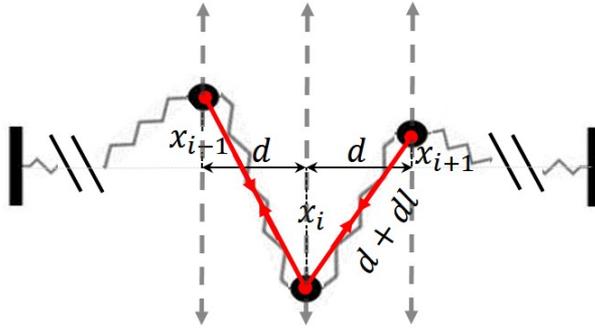


**Abb. 4.6:** Schema des Aktuator-Systems in Ruhelage als gekoppelter gedämpfter getriebener Oszillator. Die Größe  $d$  bezeichnet den Abstand der horizontal gelagerten Balken. Die Auslenkung in vertikale Richtung ist jeweils mit  $x$  angegeben.

Die in Gleichung 4.5 dargestellte Rückstellkraft kann anhand Abb. 4.7 abgeleitet werden [31]. Dabei lässt sich die Dehnung der Aluminiumoberfläche  $dl$  aus dem Aktuator-Abstand  $d$  und der Auslenkung der Aktuatoren nach Pythagoras bestimmen.  $E$  ist die Elastizitätskonstante von Aluminium.  $Q$  beschreibt die Querschnittsfläche der gedehnten Aluminiumoberfläche.

$$F_i^r(x_i, x_{i-1}, x_{i+1}) = -\frac{(E \cdot Q)}{d} \cdot \left[ \left( 1 - \frac{1}{\sqrt{1 + \frac{(x_i - x_{i-1})^2}{d^2}}} \right) \cdot (x_i - x_{i-1}) + \left( 1 - \frac{1}{\sqrt{1 + \frac{(x_i - x_{i+1})^2}{d^2}}} \right) \cdot (x_i - x_{i+1}) \right] \quad (4.5)$$

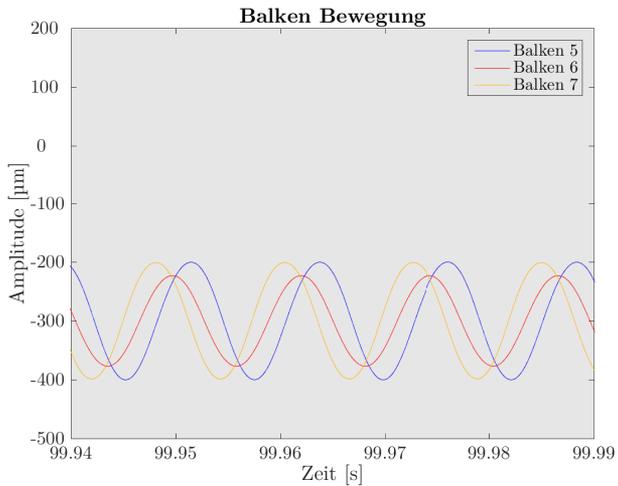
Eine Simulation dieses Modells unter Berücksichtigung der Materialkonstanten (siehe Kapitel 4.1) zeigt die Charakteristik des Systems (Abb. 4.8). Bei der Annahme eines hohen Reibungskoeffizienten ( $r_i \gg \frac{E \cdot Q}{d}$ ) kann die sinusförmige Bewegung bei phasenverschobener Ansteuerung beobachtet werden



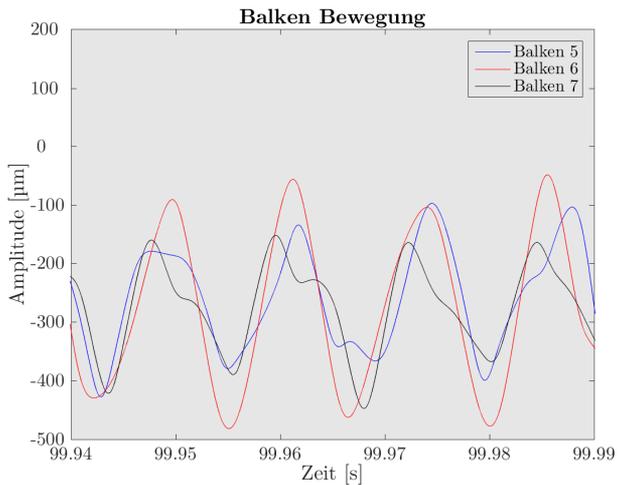
**Abb. 4.7:** Schema der bewegten Aktuatoren zur Ableitung der Rückstellkraft. Die Dehnung der Aluminiumoberfläche  $dl$  lässt sich aus dem Aktuator-Abstand  $d$  und der Auslenkung der Aktuatoren  $x$  nach Pythagoras berechnen. Die Kraft in Richtung des Blechs ist durch die elastische Verformung des Aluminiums bestimmt und teilt sich in eine horizontale und die zu berechnende vertikale Kraftkomponente auf.

(siehe Abb. 4.8a). Bei geringen Amplituden ( $A \leq 100 \mu\text{m}$ ) folgen die einzelnen Aktuator-Balken dem gewünschten Ansteuerungssignal mit einer geringen Abweichung. Für größere Amplituden (z. B.  $250 \mu\text{m}$ ) zeigt das System eine irreguläre Dynamik, die durch die starke nichtlineare Kopplung (siehe Gleichung 4.5) begründet ist. Dies ist analog bei einem kleinen Reibungskoeffizienten zu beobachten (siehe Abb.4.8b).

Das Systemverhalten zeigt deutlich, dass eine Regelung der Aktuator-Balken insbesondere für große Amplituden notwendig ist. Dabei muss nicht nur sichergestellt werden, dass die entsprechende Signalform auf die Bewegung des Balkens übertragen wird, sondern auch die entsprechende Phasendifferenz zwischen den einzelnen Aktuator-Balken muss korrekt umgesetzt werden. Nur dann ist es möglich die Korrektheit der Oberflächenwellen im Rahmen der vorgegebenen Genauigkeit von fünf Prozent zu garantieren.



(a) Darstellung von drei Aktuator-Balken für eine Amplitude vom  $100 \mu\text{m}$ .



(b) Darstellung von drei Aktuator-Balken für eine Amplitude vom  $250 \mu\text{m}$ .

**Abb. 4.8:** Darstellung der unregelmäßigen Bewegung der drei mittleren Aktuator-Balken bei einer Amplitude von  $100 \mu\text{m}$  (links) und  $250 \mu\text{m}$  (rechts im unregelmäßigen Fall). Für kleine Amplituden folgt das System nahezu dem vorgegebenen Signal, für hohe Amplituden zeigt die Balkenbewegung eine irreguläre Dynamik.

## 4.4 Regelungskonzept: Iterativ Lernende Regelung

Im Folgenden wird ein Ansatz zur Regelung der Aktuator-Bewegung und somit zur Regelung der Oberflächenwelle vorgestellt. Die Stellgröße zur Ansteuerung beziehungsweise Regelung des Aktuator-Systems ist für jeden Balken jeweils ein Spannungswert, dieser ist proportional zu der auf den Balken wirkenden Lorentzkraft. Am System lässt sich die Balkenbewegung im betrachteten Bereich exakt verfolgen und zur Regelung der Trajektorie verwenden. Die Betrachtung der nichtlinearen Kopplung als zeitlich variante Störung für jeden Balken mit einer periodischen Ansteuerung lässt den Einsatz von konventionellen Reglern für die Erzielung der vorgegebenen Genauigkeit als nicht sinnvoll erscheinen [25].

*Iterativ Lernende Regelungen* (ILR, engl.: ILC) hingegen werden eingesetzt um das Verhalten von wiederkehrenden Prozessen durch Lernen von vorherigen Iterationen zu verbessern. Diese Strategie ermöglicht die kontinuierliche Verbesserung der Ansteuerung in einem dynamischen System. Die Idee geht auf die Arbeit von Arimoto et al. [7] für den Einsatz in der Robotik zurück.

Die Anforderungen für den Einsatz einer Iterativ Lernenden Regelung sind in [3] beschrieben. Das Aktuator-System erfüllt diese, da folgende Randbedingungen für das System gelten:

1. Verwendung einer periodischen (hier: sinusförmigen) Ansteuerung
2. Festlegung des Nulldurchgangs als einheitlichen Startpunkt für jede Periode
3. Stabilisierung des Systems bei Ansteuerung mit einem beliebigen Signal

### 4.4.1 Grundlagen zur Iterativ Lernenden Regelung

Es existieren verschiedene Darstellungsformen für die Iterativ Lernende Regelung. Die folgende Darstellung (Gleichung 4.6) orientiert sich an dem Ansatz von Arimoto [7]. Eine neue Ansteuerungstrajektorie  $u_{j+1}$  wird aus der vorherigen Trajektorie  $u_j$  und einem gewichteten Fehler zwischen der gewünschten Trajektorie  $y_d$  und der gemessenen Trajektorie  $y_j$  berechnet. Die Abweichung

der gemessenen von der gewünschten Trajektorie wird mit der sogenannten Lernmatrix  $\mathbf{L}$  gewichtet.

$$u_{j+1}(t) = u_j(t) + \mathbf{L}e(t), \quad e(t) = y_d(t) - y_j(t) \quad (4.6)$$

Als Starttrajektorie  $u_0$  dient die gewünschte Trajektorie  $y_d$ . Die angegebene Iterationsvorschrift konvergiert für ein System mit der Zustandsraumdarstellung  $(\mathbf{A}, \mathbf{B}, \mathbf{C})$ , wenn die induzierte Operatornorm die folgende Regel erfüllt [58]:

$$\|I - \mathbf{CBL}\|_i < 1 \quad (4.7)$$

Ein allgemeinerer Ansatz von Rockel et al. [71] ist in Gleichung 4.8 beschrieben. Dieser beinhaltet einen Regulierungsfilter  $\mathbf{S}$  für die Ansteuerungstrajektorie, einen Lernoperator  $\mathbf{\Gamma}$  mit einem Messfilter  $\mathbf{M}$ , der auf das gemessene Signal angewandt wird. Zudem ist ein Vorfilter  $\mathbf{P}$  für die vorgegebene Trajektorie vorgesehen.

$$u_{j+1}(t) = \mathbf{S}u_j(t) + \mathbf{\Gamma M}y_j(t) - \mathbf{P}y_d(t) \quad (4.8)$$

Ein gängiger Ansatz (Gleichung 4.9) kann aus Gleichung 4.8 mit  $\mathbf{M} = \mathbf{I}$ ,  $\mathbf{P} = \mathbf{\Gamma}$  und der zusätzlichen Filtermatrix  $\mathbf{Q}$  konstruiert werden.

$$u_{j+1}(t) = \mathbf{Q}[u_j(t) + \mathbf{L}(y_j(t) - y_d(t))], \quad \mathbf{L}, \mathbf{Q} \in \mathbb{R}^{n \times n} \quad (4.9)$$

Es gibt verschiedene Reglertypen für unterschiedliche Anwendungen von lernenden Reglern. Darunter fallen  $\mathcal{H}_\infty$ -Methoden, quadratisch optimale ILR, Regler höherer Ordnung, inversionsbasierte Regler und P-Typ beziehungsweise PD-Typ Iterativ Lernende Regler. Detaillierte Beschreibungen dieser Typen sind in [14], [54], [20] und [19] wiedergegeben.

### 4.4.2 Konvergenz, Design und Implementierung

Zunächst wird die Konvergenz der allgemeinen ILR Vorschrift (4.9) betrachtet und dann ein PD-Typ ILR für die Regelung des Aktuator-Systems beschrieben.

Analog zu Bristow et al. [14] wird die „lifted system“-Darstellung in Gleichung 4.10 und die Iterationsvorschrift aus Gleichung 4.9 verwendet. Dabei beschreibt  $y_0$  den System Offset,  $\mathbf{G}$  die Systemmatrix, die das Verhalten des Systems beschreibt, und  $u_j$  die Ansteuerungstrajektorie.

$$y_j = y_0 + \mathbf{G}u_j \quad (4.10)$$

Mit dem Fehler  $e_j = y_d - y_j$  ergibt sich die Gleichung 4.11.

$$e_j = y_d - \mathbf{G}u_j - y_0 \quad (4.11)$$

Die Iterationsvorschrift wird dann beschrieben durch Gleichung 4.12 mit der  $\mathbf{Q}$ -Filtermatrix, der Einheitsmatrix  $\mathbf{I}$  und dem Lernoperator  $\mathbf{L}$ .

$$u_{j+1} = \mathbf{Q}(\mathbf{I} - \mathbf{L}\mathbf{G})u_j + \mathbf{Q}\mathbf{L}(y_d - y_0) \quad (4.12)$$

Der Spektralradius ist gegeben durch  $\lambda(\mathbf{A}) = \max_i |\lambda_i(\mathbf{A})|$ , dabei ist  $\lambda_i(\mathbf{A})$  der  $i$ -te Eigenwert einer gegebenen Matrix  $\mathbf{A}$ .

$$|\lambda_i(\mathbf{Q}(\mathbf{I} - \mathbf{L}\mathbf{G}))| < 1 \quad \forall i \quad (4.13)$$

Gleichung 4.13 beschreibt eine notwendige und hinreichende Bedingung für die Konvergenz des Fehlers [54], wenn  $\mathbf{G}$ ,  $\mathbf{Q}$  und  $\mathbf{L}$  kausale Filter darstellen. Das heißt, es sind untere Dreiecksmatrizen mit Toeplitz-Form, also einheitlichen Einträgen auf den Diagonalen. Es kann gezeigt werden, dass dies eine Stabilitätsgrenze für lineare ILR-Gesetze wie (4.6) definiert [54].

Gleichung 4.14 folgt durch Kombination von (4.9) mit (4.11).

$$e_{j+1} = (\mathbf{I} - \mathbf{G}\mathbf{Q}\mathbf{L})e_j + \mathbf{G}(\mathbf{I} - \mathbf{Q})u_j \quad (4.14)$$

Durch Einsetzen von  $e_{j+1} = y_d - y_{j+1}$  und einer Fixpunktmodifikation von (4.12) in Gleichung 4.14 ergibt sich

$$e_\infty - e_{j+1} = \mathbf{G}\mathbf{Q}(\mathbf{I} - \mathbf{L}\mathbf{G})\mathbf{G}^{-1}(e_\infty - e_j) \quad (4.15)$$

$$\|e_\infty - e_{j+1}\|_2 < \gamma \|e_\infty - e_j\|_2 \quad (4.16)$$

Die Bedingung für monotone Konvergenz (4.16) kann mithilfe von Gleichung 4.17 erreicht werden [66][14]. Die Abbildung  $\bar{\rho}$  bezeichnet die Spektralnorm, also den maximalen Singulärwert.

$$\gamma = \bar{\rho}(\mathbf{G}\mathbf{Q}(\mathbf{I} - \mathbf{L}\mathbf{G})\mathbf{G}^{-1}) < 1 \quad (4.17)$$

Für  $\mathbf{L} = \mathbf{G}^{-1}$  ist  $\gamma = 0$  und  $e_\infty = 0$  wird in einem Iterationsschritt erreicht. Durch Invertieren der Systemmatrix kann die Trajektorie zur Eliminierung der Regelabweichung berechnet werden. Bei der Invertierung kann in der Regel die Systemmatrix  $\mathbf{G}$  nur durch eine Matrix  $\hat{\mathbf{G}}$  angenähert werden. Die Anwendung von  $\mathbf{L} = \hat{\mathbf{G}}^{-1}$  kann verwendet werden, wenn  $\hat{\mathbf{G}}$  ein geeignetes Modell beschreibt [53]. Dieser Ansatz reagiert jedoch empfindlich auf Modellunsicherheiten [14].

Nach dem Vorschlag von Moore [58] wird für das beschriebene Modell des Aktuator-Systems mit starker nichtlinearer Kopplung ein PD-Typ ILR gewählt. Dabei teilt sich die Matrix  $\mathbf{L}$  in einen P-Anteil  $k_p\mathbf{I}$  und einen D-Anteil  $k_d\mathbf{D}$  auf (siehe Gleichung 4.18). Zur Erhöhung der Robustheit und zur Unterdrückung von Rauschen kann ein zusätzlicher  $\mathbf{Q}$ -Filter eingesetzt werden.

$$u_{j+1} = \mathbf{Q}(u_j + \underbrace{(k_p\mathbf{I} + k_d\mathbf{D})}_{\mathbf{L}} e_j), \quad e(t) = y_d(t) - y_j(t) \quad (4.18)$$

Für die Betrachtung eines diskreten System mit einer Totzeit  $d$  und einem diskreten PD-Typ ILR mit  $\mathbf{Q} = \mathbf{I}$  ist der  $k$ -te Abtastwert der Ansteuerungstrajektorie zum Zeitpunkt  $j+1$  analog zu [19] in Gleichung 4.19 dargestellt.

$$u_{j+1}(k) = u_j(k) + k_p e_j(k+d) + k_d(e_j(k+d+1) - e_j(k+d-1)) \quad (4.19)$$

Für  $k_d = 0$  ergibt dies einen P-Typ ILR. Im Folgenden wird einer der aktuellen Arbeitspunkte aus den Windkanalversuchen betrachtet. Die Wellenparameter sind dabei  $250\ \mu\text{m}$  Amplitude, 81 Hz Frequenz und 160 mm Wellenlänge, dies entspricht einer Phasenverschiebung von 45 Grad bei einem Balkenabstand von 20 mm [52]. Es wird ein System aus zehn Balken simuliert, wobei der Fokus auf die mittleren Balken gelegt wird. Die vorgegebene maximale Regelabweichung von der Soll-Trajektorie beträgt wie bereits beschrieben fünf Prozent.

Die in Bristow et al. [14] gezeigte Konfiguration von Iterativ Lernenden Reglern wird hier verwendet. Es werden verschiedene Kombinationen von Verstärkungsfaktoren getestet, um sich einer zufriedenstellenden Regler-Konfiguration anzunähern.

$$\mathbf{D} = \begin{pmatrix} -1 & 1 & 0 & \dots & 0 \\ -0,5 & 0 & 0,5 & \ddots & \vdots \\ 0 & \ddots & \ddots & \ddots & 0 \\ \vdots & \ddots & -0,5 & 0 & 0,5 \\ 0 & \dots & \dots & -1 & 1 \end{pmatrix}$$

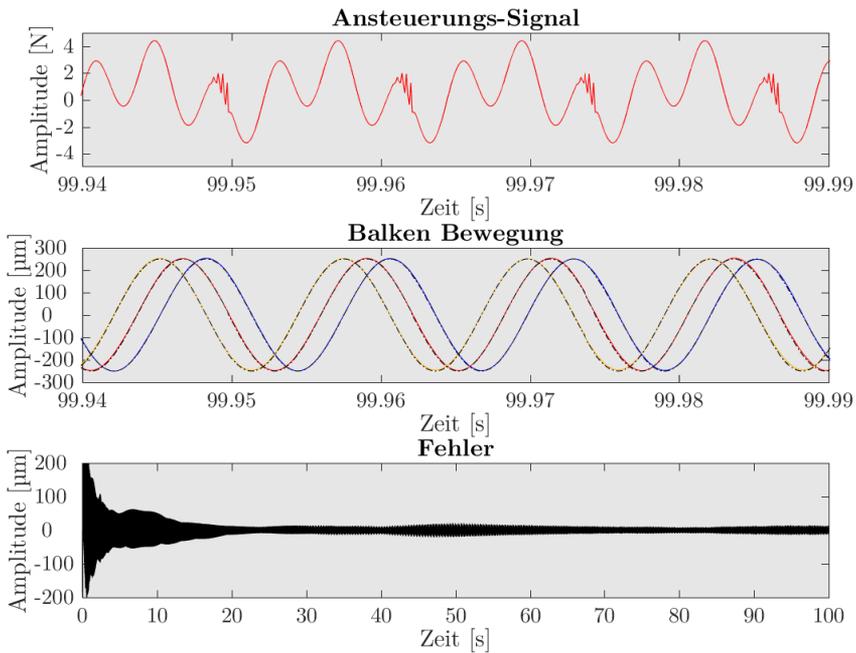
Die Ableitungsmatrix  $\mathbf{D}$  bildet einen zentralen Differenzenquotienten ab. In der vorliegenden Anwendung wird  $\mathbf{Q} = \mathbf{I}$  gesetzt. Für hohe Verstärkungsfaktoren  $k_p$  und  $k_d$  ist eine schnelle Konvergenz, aber eine zunehmende Instabilität zu beobachten. Geringe Verstärkungsfaktoren führen erst nach längerer Simulationszeit in einen stabilen Zustand.

Mit den Verstärkungsfaktoren  $k_p = 0,075$  und  $k_d = 4$  werden für die Reduktion der Regelabweichung unter  $\pm 50 \mu\text{m}$  etwa zehn Sekunden benötigt. In Abb. 4.9 ist die Ansteuerungstrajektorie und die Bewegung der Balken dargestellt sowie die Abweichung zwischen der geforderten und der resultierenden Balkenbewegung. Die gestrichelte Linie im mittleren Abschnitt beschreibt die gewünschte Trajektorie. Der Fehler oszilliert nach 20 s in einem beschränkten Bereich, sodass keine monotone Konvergenz beobachtet werden kann. Das System ist jedoch für die Zeit der Simulation stabilisiert.

**Tabelle 4.1:** Verschiedene Konfigurationen des PD-Typ ILR werden für die Regelung der gekoppelten Aktuator-Balken verwendet. Kleine Verstärkungsfaktoren ohne Schwellwert zeigen nach 100 s die gleiche Abweichung wie große Verstärkungsfaktoren mit Schwellwert, bei dessen Unterschreitung der Regler ausgeschaltet wird. Regler mit großen Verstärkungsfaktoren konvergieren deutlich schneller.

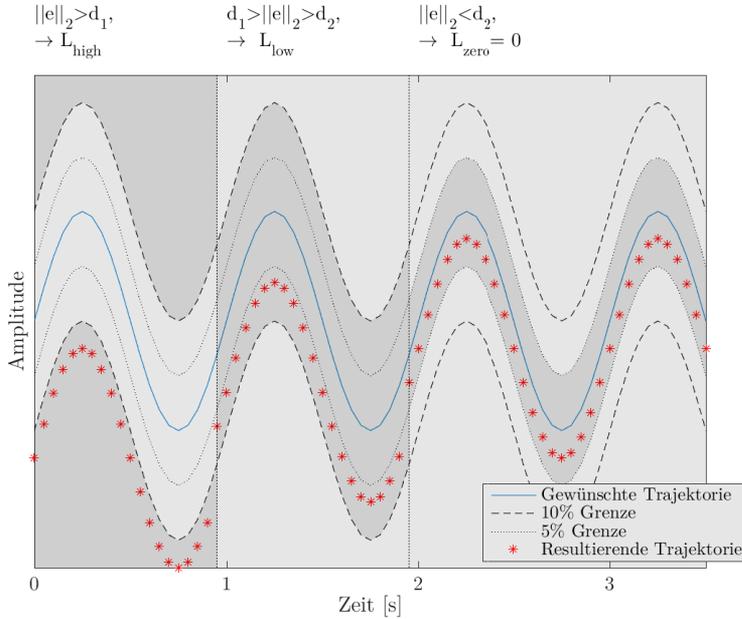
Verst.-Faktoren	Schwellwert	Konv.-Zeit [s]	Abw. [%]
$k_p = 0,075, k_d = 4$		20	5,64
$k_p = 0,075, k_d = 4$	$\ e_i\ _2 > 0,05$	20	4,00
$k_p = 1,1, k_d = 50$	$\ e_i\ _2 > 0,10$	2,5	5,64

In Tabelle 4.1 sind drei verschiedene Simulationen aufgeführt. Dabei werden die Verstärkungsfaktoren bei einer Regelabweichung unterhalb des gegebenen



**Abb. 4.9:** Der Regler erzeugt für jeden der gekoppelten Aktuator-Balken eine andere Ansteuerungs-Trajektorie. Diese ermöglichen die Generierung von phasenverschobenen sinusförmigen Bewegungen zur Erzeugung wandernder transversaler Oberflächenwellen. Der Fehler, also die Abweichung des gemessenen Signals von dem als gestrichelte Linie dargestellten gewünschten Bewegung, wird innerhalb der ersten 20 s reduziert, oszilliert danach aber in einem beschränkten Bereich.

Schwellwertes auf  $k_p = 0$  und  $k_d = 0$  gesetzt. Damit lässt sich sowohl die resultierende Regelabweichung bei den niedrigen Verstärkungsfaktoren reduzieren, als auch ein stabiler Zustand bei höheren Verstärkungsfaktoren realisieren.



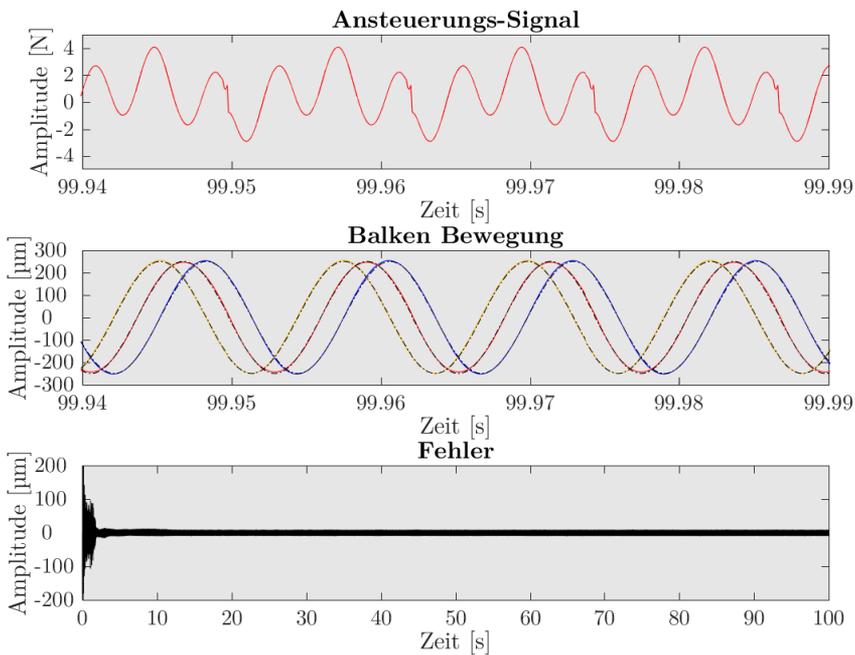
**Abb. 4.10:** Die eingesetzte Iterativ Lernende Regelung passt die Verstärkungsfaktoren über die Lernmatrix  $\mathbf{L}$  abhängig von der Regelabweichung an.

Nach dem Prinzip eines Mehrpunktreglers lassen sich diese Regler nun zu einem Regler mit variablen Verstärkungsfaktoren zusammenführen. Die genaue Funktionsweise des Reglers ist in Abb. 4.10 dargestellt.

Für zwei gewählte Schwellwerte  $d_1$  und  $d_2$ , wobei  $d_2$  die maximal zulässige Regelabweichung darstellt, wird in jeder Iteration entschieden, welche Lernmatrix  $\mathbf{L}$  verwendet wird. Ist die genormte Regelabweichung  $\|e\|_2 > d_1$ , dann wird eine Matrix mit hohen Verstärkungsfaktoren  $L_{high}$  gewählt. Für  $d_1 > \|e\|_2 > d_1$  wird eine Matrix mit niedrigen Verstärkungsfaktoren gewählt. Unterschreitet  $\|e\|_2$  den Schwellwert  $d_2$ , so wird der Regler ausgeschaltet.

### 4.4.3 Simulationsergebnisse

Für den Iterativ Lernenden Regler mit variablen Verstärkungsfaktoren werden die beiden Regler mit Schwellwerten aus Tabelle 4.1 zusammengeführt. Das Ergebnis der Simulation ist in Tabelle 4.2 dargestellt. Eine Regelabweichung von weniger als 0,01 mm wird nach 2,5 s erreicht. Durch die Abschaltung des Reglers unterhalb des Schwellwertes von  $\|e_i\|_2 > 0,05$  bleibt die Regelabweichung nahezu konstant (siehe Abb. 4.11).



**Abb. 4.11:** Die Anwendung des Iterativ Lernenden Reglers mit variablen Verstärkungsfaktoren ermöglicht die Reduktion des Fehlers, also der Abweichung des gemessenen Signals von dem als gestrichelte Linie dargestellten gewünschten Bewegung. Nach 2,5 s ist das Signal in einen stabilen Zustand überführt.

Der Regler ist gegenüber geringen Änderungen der Wellenparameter robust. Beispielsweise verringert sich die Performanz bei den Wellenparametern

**Tabelle 4.2:** Die Iterativ Lernende Regelung wird analog zu einem Mehrpunktregler je nach Regelabweichung durch die Verstärkungsfaktoren angepasst. Damit kann eine geringe Konvergenzzeit gegenüber einer Konfiguration ohne Schwellwerte erzielt und die Abweichung stabilisiert werden.

Verst.-Faktoren	Schwellwert	Konv.-Zeit [s]	Abw. [%]
$k_p = 1,1, k_d = 50$	$\ e_i\ _2 > 0,10$	2,5	3,88
$k_p = 0,075, k_d = 4$	$\ e_i\ _2 > 0,05$		

260  $\mu\text{m}$ , 82 Hz und 46 Grad Phasenverschiebung. Es stellt sich ein stabiler Zustand nach etwa 17 s ein, jedoch steigt die Abweichung von der gewünschten Trajektorie geringfügig über fünf Prozent. Für den gesamten Parameterbereich ( $F < 200$  Hz,  $A < 1$  mm,  $\lambda \geq 120$  mm) ist der Regler jedoch in dieser Form nicht einsetzbar.

Die hier vorgestellte Regelung kann innerhalb der modular aufgebauten Aktuator-Ansteuerung in das Modul *Datenverarbeitung* integriert werden. Für den Einsatz im Windkanal mit der externen Strömungsregelung ist die weitere Betrachtung hinsichtlich der Robustheit der Regelung notwendig. Der Regler muss für den flexiblen Einsatz und die Realisierung großer Amplituden robust gegenüber allen Parameteränderungen (Amplitude, Frequenz, Phase, Offset, Reibung) sein.

## 4.5 Zusammenfassung

Das Aktuator-System bewegt mithilfe in Magnetfeldern gelagerter auf PCB-Trägern verklebter Luftspulen eine Aluminiumoberfläche und erzeugt so transversale Oberflächenwellen. Das bisher verwendete Aktuator-System stellt die pro Aktuator benötigte Kraft von 20 N für die Wellenlänge  $\lambda = 160$  mm und die Amplitude  $A = 1000 \mu\text{m}$  auf einer Aluminiumoberfläche mit einer Stärke von 0,3 mm zur Verfügung. Für den Einsatz im Windkanal und insbesondere für die Verwendung der externen Strömungsregelung ist eine echtzeitige Variation der Wellenparameter in der Aktuator-Ansteuerung implementiert. Diese generiert analoge Signale auf einem Echtzeitsystem der

Firma National Instruments. Die Aktuierung wird durch eine nichtlineare Kopplung zwischen den einzelnen Aktuator-Balken beeinflusst. Die notwendige Genauigkeit der Aktuierung wird in einem Bereich von fünf Prozent abweichend vom geforderten Ansteuerungssignal für die Bewegung der einzelnen Balken angenommen. Dazu wurde eine Iterativ Lernende Regelung mit variablen Verstärkungsfaktoren in einer Simulation realisiert, die dieser Beschränkung im Arbeitspunkt  $250\ \mu\text{m}$  Amplitude, 81 Hz Frequenz und 160 mm Wellenlänge genügt.



# 5 Modellvalidierung und Model-in-the-Loop-Simulation

Dieses Kapitel befasst sich mit der Validierung der Netzwerkkomponenten, des Netzwerkes als Gesamtkonzept sowie der Ankopplung des Netzwerkmodells an die externe Regelung und die Aktuator-Hardware. Für die Validierung muss in der Regel die in Kapitel 3 beschriebene Modellimplementierung herangezogen werden. Die Inhalte dieses Kapitels finden sich zu unterschiedlichen Teilen in diesen Veröffentlichungen von Dück et al.: [27], [29], [28], [31], [30].

## 5.1 Mikrocontrollerbasiertes Testnetzwerk

Für die modellbasierte Entwicklung eines mikrocontrollerbasierten räumlich ausgedehnten Aktuator- und Sensornetzwerkes ist es wichtig möglichst realitätsnahe Netzwerkparameter zu verwenden. Dazu wird hier ein kleiner Simulationsansatz zur Validierung von wesentlichen Parametern mithilfe einer physikalischen Repräsentation des Modells verwendet. Die modellbasierte Netzwerkkonfiguration in MATLAB/Simulink und TrueTime wird damit an die realen Gegebenheiten angepasst. Als Plattform lassen sich dafür verschiedene Architekturen verwenden.

Als physikalische Netzwerkteilnehmer stehen sowohl die Standard-PC-Architektur als eingebettetes System [51] als auch verschiedene mikrocontrollerbasierte Plattformen zur Verfügung. Für die spätere Realisierung des Netzwerkes soll die iNODE-Plattform verwendet werden. Diese zeichnet sich durch Flexibilität und Energieeffizienz aus [75]. Für die Evaluierung der Kommunikationsparameter wird jedoch der Raspberry Pi verwendet. Diese Plattform ist seit 2012 erhältlich und hat sich in Lehre und Forschung etabliert. Sie ermöglicht insbesondere eine einfache Programmierung innerhalb des Linux-Betriebssystems [70].

Beim Aufbau der Netzwerk-Testumgebung wird eine minimale Konfiguration mit jeweils einem der vier möglichen Knoten  $NCAP_I$ ,  $NCAP_C$ ,  $STM_A$ ,  $STM_S$  (siehe Abb. 5.1) verwendet. Es wird ein switched Ethernet Protokoll mit einer sternförmigen Vernetzung aufgebaut. Dabei werden vier *Raspberry Pi B* miteinander mithilfe eines *Allied Telesis Switch* [5] und jeweils einem 30 cm *Cat.5e* Kabel in einem konventionellen LAN verbunden.

Das Verhalten der Testumgebung wird mit verschiedenen Übertragungsparametern für die Modellierung charakterisiert und mit der Simulation abgeglichen. Die Parameter werden über Annahmen und Tests identifiziert. Entsprechende Testmessungen, vergleichbar zu den Simulationsergebnissen, sind die Basis zum Vergleich von Testumgebung und Modell. Diese Vorgehensweise wird sowohl für die unterste Kommunikationsebene als auch für die Anwendungsschicht appliziert. Im Speziellen dient die Sendedauer im Netzwerk beziehungsweise die Übertragungsdauer in der entsprechenden Netzwerkschicht als Grundlage zum Vergleich. Sofern die Ergebnisse der Testmessungen mit dem Modell in TrueTime reproduziert werden können, sind die Modellparameter validiert und von den Simulationsergebnissen hochskalierter Modelle können Aussagen für ein reales Netzwerk abgeleitet werden.



**Abb. 5.1:** Die vier Raspberry Pi B Knoten werden über ein switched Ethernet Netzwerk verbunden. Die Kommunikationsereignisse innerhalb des Netzwerkes werden zur Parameterevaluation überwacht.

Für die Untersuchung von Aktuator- und Sensornetzwerken mit einer großen Anzahl an Knoten können die validierten Parameter für die Modellierung und Simulation verwendet werden. Für die Applikation der Strömungsregelung muss die Taktung einer externen Regelung in die Modellbildung einfließen.

## 5.2 Validierung der Kommunikation

Für die Charakterisierung eines Netzwerkes ist die Auswahl eines Protokolls, einer daraus resultierenden minimalen und maximalen Paketgröße sowie einer maximalen Bandbreite notwendig. Die angesetzte Bandbreite sollte allerdings im Netzwerkbetrieb nicht vollständig ausgeschöpft werden.

Das Verhalten der Kommunikation in der Simulation des echtzeitigen Aktuator- und Sensornetzwerkes wird maßgeblich durch die Wahl der Wertebereiche der Parameter bestimmt. Verschiedene zu wählenden Größen lassen sich durch einfache Annahmen einschränken. So ist die Übertragungsrate in einem Fast Ethernet auf 12,5 MB/s beschränkt. Für den Paketverlust kann als Daumenregel ein Limit von fünf Prozent in einem geschlossenen Netzwerk angenommen werden. Höhere Paketverlustraten deuten auf ein Problem mit der Infrastruktur hin, wobei Paketverluste auf allen Netzwerkschichten auftreten können und sich dabei gegenseitig beeinflussen [27]. In der Regel treten Paketverluste bei Kollisionen auf. Die Verwendung eines switched Ethernet Netzwerkes verhindert Paketkollisionen durch den Pufferspeicher im Switch. Dies bringt aber eine zusätzliche Hardwarebeschränkung mit sich. Beim Einsatz des hier verwendeten Switches der Firma Allied Telesis [5] existiert ein Datenpuffer von 256 Kilobytes. Es ergibt sich damit eine Beschränkung auf maximal 14 Knoten (ein Master und 13 Slaves) in einem Netzwerk, da beim Anmeldeprozess jeder Slave-Knoten einen Broadcast auf das Netz sendet. Die Beschränkung der Knotenanzahl verhindert einen Pufferüberlauf und vermeidet so das Auftreten von Paketverlusten. Für ein Netzwerk mit 14 Knoten ergeben sich bei der Anmeldung  $13 \times 13$  Nachrichten mit jeweils 1500 Bytes, also etwa 248 Kilobytes, die gleichzeitig durch den Switch verteilt werden müssen. Bei 15 Knoten mit entsprechend 14 Broadcasts sind es bereits etwa 287 Kilobytes.

Zur Bestimmung von Paketverlusten schlagen Ionete et al. [45] eine Analyse des Netzwerkes mit echtzeitfähigen Maschinen vor. Die Raspberry Pi Testumgebung wird auf ähnliche Weise verwendet, um verschiedene Parameter

zu validieren. Tobagi et al. [84] bestimmen abhängig von der Ende-zu-Ende-Verzögerung und den Paketverlusten die Anzahl der maximalen Datenströme, woraus die Bitrate abgeleitet werden kann. Diese ist im Bereich von 43% (5,375 MB/s) bis 86% (10,75 MB/s) angegeben.

### 5.2.1 Validierung der Netzwerkschicht

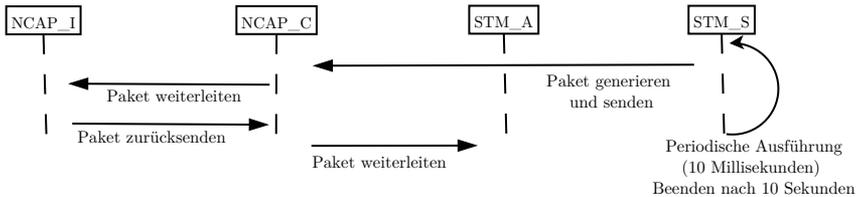
Die Parameter der Netzwerkschicht werden durch verschiedene Methoden bestimmt. Somit ist die maximale Anzahl der Knoten durch die Hardwarebeschränkung des Switches [5] bereits zu Beginn festgelegt. Durch die Größe des Switch-Puffers ist die Anzahl der Teilnehmer auf 14 Knoten beschränkt. Die Testumgebung stellt eine physikalische Repräsentation eines möglichen Modells dar. Diese wird im Modell durch vier Knoten (NCAP<sub>I</sub>, NCAP<sub>C</sub>, STM<sub>A</sub>, STM<sub>S</sub>) und einen Netzwerk-Block der TrueTime-Toolbox repräsentiert. Als MAC-Protocol wird switched Ethernet gewählt. Dieses wird sowohl im Simulink Modell als auch in der Raspberry Pi Testumgebung verwendet. Zur Realisierung eines einfachen und schnellen Datenaustausches ist die Nachrichtenlänge auf die Ethernet-MTU mit 1500 Bytes begrenzt. Damit sind keine zusätzlichen Betrachtungen von Segmentierung, Flusskontrolle und Staukontrolle notwendig. Zudem können Paketverluste ausgeschlossen werden.

**Tabelle 5.1:** Evaluerte Bandbreitenparameter basierend auf Messungen mit verschiedenen Paketgrößen mit dem UDP-Protokoll durchgeführt mit dem Unix-Standardwerkzeug „netcat“.

Fall	Wert	Prozentsatz
schlechtester	4,9 MB/s	39 %
bester	7,1 MB/s	57 %
Durchschnitt	6,6 MB/s	52 %

Die Bandbreite wird mit dem Unix Standardwerkzeug „netcat“ erfasst. Die durchschnittliche Bandbreite lässt sich dabei mit 6,6 MB/s angeben (siehe Tabelle 5.1). Ein Vergleich zwischen Modell und physikalischer Testumgebung lässt sich mit einem vereinfachten Kommunikationsparadigma mit Anwendungsbezug durchführen. Eine mehrfache Parameteränderung der Aktuatorik als Reaktion auf eingehende Messwerte dient als Beispielanwendung. Der

Kommunikationsablauf ist im Sequenzdiagramm in Abb. 5.2 dargestellt. Ausgehend vom Sensor werden Daten durch das Netzwerk gesendet. In einem periodischen Prozess werden 1000 Pakete in zehn Sekunden bearbeitet. Im Zyklus von 9,9 ms sendet der  $STM_S$  ein Paket zum  $NCAP_C$ , dieser sendet das Paket weiter zum  $NCAP_I$ . Das Paket wird von dort zurück zum  $NCAP_C$  und schließlich zum  $STM_A$  übertragen.



**Abb. 5.2:** Die Kommunikationssequenz des Paketes durch die vier vernetzten Knoten. Diese Sequenz wird in einem Intervall von circa zehn Millisekunden ausgehend vom  $STM_S$  initiiert, sodass innerhalb von zehn Sekunden etwa 4000 Kommunikationsereignisse pro Experiment beobachtet werden.

Da hier die MAC-Parameter evaluiert werden, erfolgt die Kommunikation nur auf der untersten Netzwerkebene. Es werden keine zusätzlichen Netzwerkschichten betrachtet. Für den Vergleich von Modell und Testumgebung ist das oben genannte Kommunikationsparadigma auf der Raspberry Pi Plattform implementiert und wird unter Einbeziehung verschiedener fester Paketgrößen durchlaufen. Abläufe mit jeweils 64 Bytes, 732 Bytes und 1400 Bytes werden evaluiert. Um sicherzustellen, dass die Pakete durch zusätzliche Daten nicht die maximale Übertragungsgröße des Ethernet-Protokolls in der Testumgebung überschreiten, wird die Parameter-Validierung mit 1400 Byte-Paketen durchgeführt. In der Simulation sind Probleme in Bezug auf eine variable Paketgröße durch Verwendung der größtmöglichen Pakete (1500 Bytes) ausgeschlossen. Für die Vergleichbarkeit der Netzwerkschicht in der Testumgebung und in der Simulation ist die Betrachtung verschiedener Paketgrößen wichtig.

Die Simulation des TrueTime-Modells liefert als Resultat die Auslastung des Netzwerkes auf der gesamten Zeitskala. Um diese Daten näherungsweise für einen Vergleich aus der physikalischen Testumgebung zu erhalten, werden die einzelnen Zeitpunkte vor und nach dem Senden eines Paketes für jeden Knoten gemessen. Die Aufnahme der Daten erfolgt in der Testumgebung für die drei verschiedenen Paketgrößen in jeweils  $n = 70$  Experimenten. Die Daten

werden auf die Zeitabstände  $x_i$  zwischen den deterministischen Netzwerkeignissen reduziert und deren Häufigkeit durch den Faktor  $p_i$  ausgedrückt. Der Vergleich des Modellverhaltens mit dem der Testumgebung erfolgt mithilfe der Berechnung der charakteristischen Werte, dem Mittelwert  $\bar{x}$  und der Standardabweichung  $s$  (siehe Gleichungen 5.1-5.4).

$$\bar{x} = \frac{\sum_{i=1}^n p_i x_i}{\sum_{i=1}^n p_i} \quad (5.1)$$

$$s_{\text{int}}^2 = \frac{1}{\sum_{i=1}^n \frac{1}{s_i^2}} \quad (5.2)$$

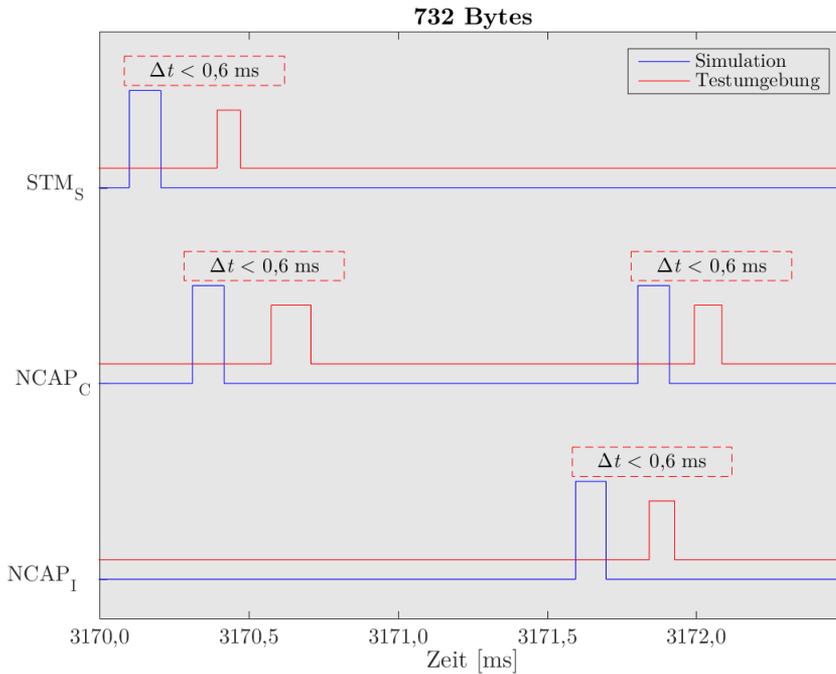
$$s_{\text{ext}}^2 = \frac{1}{n-1} \frac{\sum_{i=1}^n p_i (x_i - \bar{x})^2}{\sum_{i=1}^n p_i} \quad (5.3)$$

$$s = \max(s_{\text{int}}, s_{\text{ext}}) \quad (5.4)$$

Es ist ein periodisch funktionierender Zustandsautomat sowohl in der Testumgebung, als auch im TrueTime-Netzwerkmodell implementiert. Die Bandbreite und Paketgröße bestimmen im Modell die Übertragungszeit auf der MAC-Ebene. Die Ausführung in der Testumgebung wird mithilfe des `sleep`-Befehls verzögert. Dieser erzeugt eine Verschiebung der Messung in Abhängigkeit von der Paketgröße (Verarbeitungszeit und Wartezeit) und beeinflusst somit die Wartezeit in der Testumgebung. Die entsprechende Paketverarbeitungszeit im TrueTime-Modell wird nicht auf die Wartezeit addiert. Die zeitlichen Unterschiede sind durch eine Änderung der Wartezeit in TrueTime angepasst.

Ungeachtet des deterministischen Verhaltens des Testnetzwerkes treten Messfehler auf. Messreihen mit groben, direkt erkennbaren Messfehlern werden von vornherein ausgeschlossen, um die Analyse nicht zu verfälschen. Unsicherheiten in den Messungen, die durch zeitliche Abstände oder Abweichungen entstehen, lassen sich durch häufige Synchronisation vor jeder Messung kompensieren. Dies wird über eine initialisierende Broadcast-Nachricht ausgehend von STM<sub>A</sub> realisiert. Aufgrund der kurzen Übertragungswege sind keine nennenswerten Schwankungen beim Zeitabgleich erkennbar.

Diese Synchronisation über Broadcast-Nachrichten ist aus folgenden Gründen notwendig: Die zeitliche Synchronisation mit dem *Network Time Protocol* (NTP) resultiert in einem Versatz zwischen Messung und Simulation von etwa zehn Millisekunden nach 1000 Paketen in zehn Sekunden. Das NTP Protokoll basiert auf dem Austausch von mehreren UDP-basierten Paketen [63][56]. Das Protokoll generiert eine zusätzliche Netzwerkauslastung. Der Datenaustausch wird durch einen Hintergrundprozess gesteuert. Der Abgleich mit einer *Real-Time-Clock* (RTC) liefert ebenfalls kein zufriedenstellendes Ergebnis. Die Abstimmung in Millisekunden-Genauigkeit zwischen den einzelnen Knoten ist mit der getesteten RTC [57] nicht möglich.

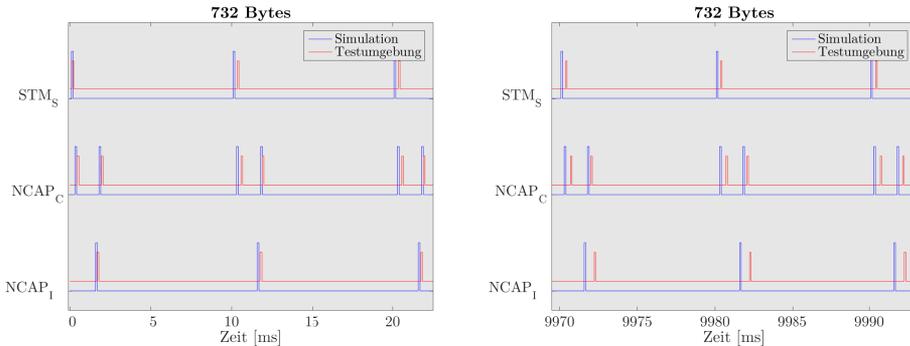


**Abb. 5.3:** Nahaufnahme für den Vergleich zwischen den Kommunikationsereignissen im Raspberry Pi basierten Testnetzwerk und der TrueTime-Simulation bei einer Paketgröße von 732 Bytes. Die Zeitunterschiede sind in diesem Beispiel im Submillisekundenbereich.

Der Vergleich von Testumgebung und Simulation erfolgt durch überlagerte

Visualisierung der Messungen und numerischen Charakterisierung der Messwerte aus der Testumgebung. Für die numerische Betrachtung werden die Wartezeiten zwischen den Ereignissen für jeden Knoten betrachtet.

Eine nähere Betrachtung zeigt eine zufriedenstellende Genauigkeit beim Vergleich, sowohl für die gewählten Netzwerkparameter als auch für die verwendete Synchronisation. Eine höhere Genauigkeit in der Testumgebung ist mit einem zusätzlichen Messkanal im Netzwerk oder über eine GPIO-Verbindung denkbar. Die Abweichungen im gezeigten Beispielausschnitt für eine Paketgröße von 732 Bytes (siehe Abb. 5.3) ist geringer als 0,5 ms. Die Abweichung über das gesamte Experiment liegt bei maximal 0,6 ms, gemittelt über alle Messungen. Für alle Paketgrößen ist eine Abweichung von weniger als einer Millisekunde zu beobachten und liegt somit innerhalb des in Kapitel 3.2 festgelegten Toleranzbereichs von drei Millisekunden.



- (a) Die ersten drei Kommunikationszyklen mit 732 Bytes Paketgröße. Simulation und Testumgebung sind zum Vergleich grafisch überlagert.
- (b) Die letzten drei Kommunikationszyklen mit 732 Bytes Paketgröße. Simulation und Testumgebung sind zum Vergleich grafisch überlagert.

**Abb. 5.4:** Resultate des Vergleichs zwischen den Kommunikationsereignissen im Raspberry Pi basierten Testnetzwerk und der TrueTime-Simulation für Pakete mit einer Größe von 732 Bytes in einem Experiment mit einer Dauer von zehn Sekunden. Die auftretenden Kommunikationsereignisse werden durch eine grafische Überlagerung am Anfang (links) und am Ende (rechts) verglichen. Die Abweichungen zwischen den Ereignissen in der Simulation und den entsprechenden Ereignissen im Testnetzwerk sind geringer als eine Millisekunde.

Die Messung des  $STM_A$ -Knotens wird nicht benötigt, da dieser nur als Datensinke fungiert. Der initiiierende Synchronisations-Broadcast wird jedoch von diesem Knoten aus gesendet. Der Vergleich von Messung und Testumgebung zeigt, dass die Messmethode genügend Präzision für die Messung der Zeitdifferenzen auf verschiedenen Knoten bietet.

Im grafischen Vergleich des TrueTime-Plots mit der Messung bei 732 Bytes (siehe Abb. 5.4) liegt ein spezielles Augenmerk auf dem Beginn (siehe Abb. 5.4a) und dem Ende (siehe Abb. 5.4b) von Simulation und Testumgebung. Dabei werden die gemessenen Netzwerkauslastungen grafisch überlagert. Die Anzahl der Netzwerkereignisse stimmt in Simulation und Messung überein und es existiert kein nennenswerter Versatz resultierend aus der Länge der Simulation (siehe Abb. 5.4b). Die einmalige Synchronisation zu Beginn ermöglicht diese hohe Genauigkeit. Falls keine zuverlässige Synchronisation verwendet wird, so ist eine Abweichung von zehn Millisekunden nach zehn Sekunden zu beobachten, sodass auch die Anzahl der Ereignisse nach Ablauf von zehn Sekunden nicht übereinstimmt [27]. Der Ablauf für Experiment und Simulation entspricht dem im Sequenzdiagramm Abb. 5.2.

In Tabelle 5.2 sind Mittelwerte und Standardabweichung für die Zeitabstände zwischen den Netzwerkereignissen und die durchschnittliche Anzahl der Ereignisse aufgelistet. Diese Verzögerung wird durch den `sleep`-Befehl erzeugt und beträgt beispielsweise für die 64 Byte-Pakete 9,99 ms. Ausgehend von einer Simulation mit einer fest vorgegebenen Zeit zwischen den Ereignissen erfolgt der Vergleich mit der Testumgebung durch graphische Überlagerung der Ereignisse. Die Auswertung der statistischen Werte zeigt eine abnehmende Anzahl von Ereignissen bei zunehmender Paketgröße. Mit der Paketgröße steigt auch der Mittelwert der Abstände zwischen den Ereignissen, die Differenzzeit hängt also von der Paketgröße ab. Es kann jedoch kein Zusammenhang zwischen der Standardabweichung und der Paketgröße oder der Anzahl der Ereignisse jedes Knotens festgestellt werden. Die Standardabweichung steigt nicht monoton mit dem zurückgelegten Weg im Netzwerk. Abweichungen in der Wartezeit zwischen den Übertragungen entstehen in der Testumgebung durch die Betrachtung verschiedener Paketgrößen. Bei günstig gewählten Wartezeiten zwischen den Ereignissen in der Simulation liegen die Abweichungen im Bereich von weniger als einer Millisekunde. Für die Bewertung der Testumgebung ist festzuhalten, dass sowohl die Übertragungszeit als auch die Wartezeit zwischen den Ereignissen abhängig von der zu verarbeitenden Paketgröße ist.

$$t_{stm_s} = 9,99 + \frac{p - 64}{64} * 0,001 \quad (5.5)$$

**Tabelle 5.2:** Charakteristische Werte der Zeitmessung in Millisekunden (Abstände zwischen den Ereignissen) zusammen mit der mittleren Anzahl Ereignisse auf dem STM<sub>S</sub>-Knoten in zehn Sekunden langen Experimenten.

	<b>64 Bytes</b>		<b>732 Bytes</b>		<b>1400 Bytes</b>	
	$\bar{x}$	s	$\bar{x}$	s	$\bar{x}$	s
STM <sub>S</sub>	9,99	0,03	10,00	0,02	10,01	0,04
NCAP <sub>C</sub>	9,99	0,05	9,99	0,05	10,00	0,07
NCAP <sub>I</sub>	9,99	0,09	10,00	0,07	10,01	0,13
NCAP <sub>C</sub>	1,08	0,09	1,40	0,07	1,73	0,11
Durchschn. Ereignisse	1001,5		1000,5		999	

Die für das Modell im vorgegebenen Kommunikationsparadigma zu verwendende Wartezeit ist in Gleichung 5.5 beschrieben. Dabei wird als Mittelwert der Wartezeit in der Testumgebung ein Wert von 9,9ms angegeben. Aufgrund des Einflusses der Paketgröße variiert die Wartezeit. Die Gleichung ist das Resultat einer linearen Anpassung an die mittleren Abstände zwischen den Sendeereignissen (siehe Tabelle 5.2). Für die Knoten STM<sub>S</sub> und NCAP<sub>I</sub> stimmen die Resultate mit den Messungen überein.

Das Ergebnis der Parameteridentifikation für die Simulation von Aktuator- und Sensornetzwerken im vorgestellten Modell ist in Tabelle 5.3 dargestellt. Für die weiteren Simulationen werden die hier validierten Parameter verwendet. Verzögerungszeiten, wie in der Beispielanwendung simuliert, müssen nicht betrachtet werden, da diese maßgeblich von der Algorithmik der externen Regelung abhängen. Die verwendete Testumgebung hat sich als eine einfache und zuverlässige Methode zur Identifikation von Netzwerkparametern in der untersten Netzwerkschicht erwiesen. Die Vergleichbarkeit von Testumgebung und Simulation zeigt, dass die evaluierten Netzwerkparameter eine sinnvolle Basis für die weitere Verwendung in räumlich ausgedehnten Aktuator- und Sensornetzwerksimulationen darstellen.

**Tabelle 5.3:** Resultierende Parameter aus der Evaluation des Testnetzwerkes für den Einsatz in der TrueTime-Netzwerksimulation.

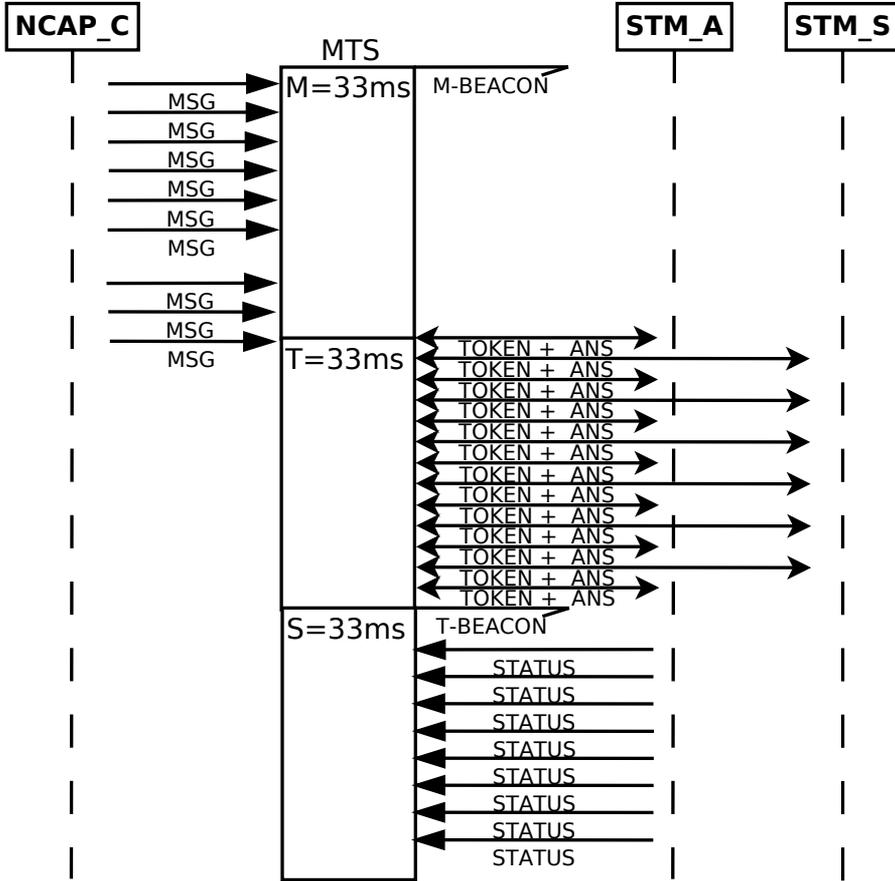
Typ	Wert
Topologie	Switched Ethernet
Bandbreite	6,6 MB/s
minimale Paketgröße	64 Bytes
maximale Paketgröße	1500 Bytes
Switch Puffergröße	256 KB
Switch Verhalten	herkömmlicher Puffer, Verwerfen bei Überlauf

### 5.2.2 Validierung der Transportschicht

Im folgenden Beispiel wird die Funktionsweise des deterministischen *MTS*-Transportprotokolls für die Verwendung in echtzeitigen Aktuator- und Sensornetzwerken anwendungsunabhängig veranschaulicht und die Echtzeitfähigkeit an einem reduzierten Modell mit drei Knoten, einem  $\text{NCAP}_C$ , einem  $\text{STM}_A$  und einem  $\text{STM}_S$  in einem Netzwerk nachgewiesen. Die Analyse und Auswertung erfolgt über die Ausgabe der *Schedule*-Daten des TrueTime-Netzwerk-Blocks.

Es wird eine Bandbreite von 1,25 MB/s verwendet. Daraus resultiert eine Übertragungszeit von 1,144 ms für ein Paket mit 1500 Bytes. Für protokollbasierte Steuernachrichten mit 64 Bytes (Beacon, Acknowledgement, Token) sind es 0,048 ms. Innerhalb der TrueTime-Toolbox wird keine zusätzliche Latenzzeit berücksichtigt. Es werden 300 ms simuliert. Die drei *MTS*-Zyklen sind äquidistant auf 100 ms aufgeteilt und dauern jeweils  $\frac{100}{3}$  ms, sodass im Experiment drei vollständige Zyklen durchlaufen werden.

Zur Validierung wurde ein Beispiel konstruiert, mit dem nachgewiesen wird, dass die Kommunikationsphasen nicht verlassen werden. Falls ein Paket nicht im aktuellen Zyklus übermittelt werden kann, so muss die Kommunikation später wieder aufgenommen werden. Es wird ebenfalls gezeigt, dass die vollständige Kommunikation bei einer verbesserten Abstimmung des Protokolls auf die Applikation in zwei statt drei Zyklen abgeschlossen ist. Der Beginn der Applikation wird von einer Wartezeit von einer Millisekunde im  $\text{NCAP}_C$



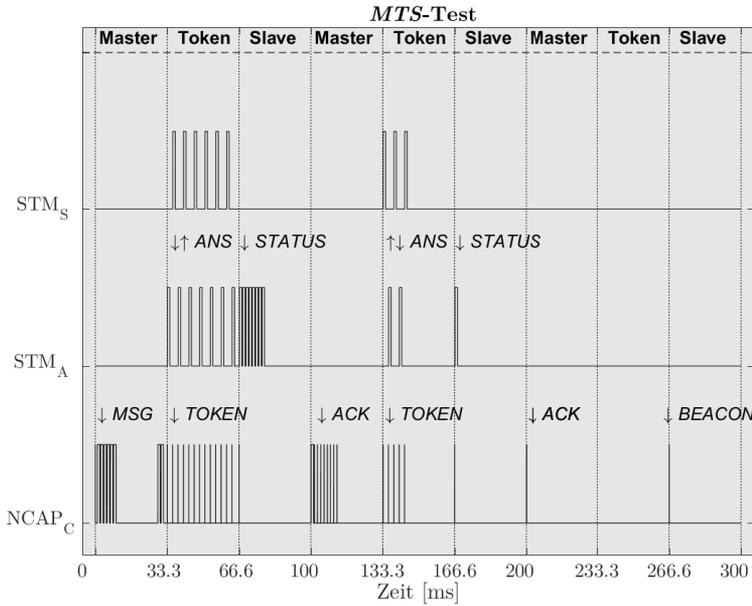
**Abb. 5.5:** Der Datenaustausch erfolgt über das *MTS*-Protokoll. Im vorgegebenen Experiment werden nicht alle Nachrichten (*MSG*) an die Transportschicht in der Master-Phase übergeben, sodass die Kommunikation dieser Nachrichten bis zum nachfolgenden Zyklus verzögert wird. Ebenso wird mit den Token und den zugehörigen Antworten verfahren.

markiert, um das *MTS*-Protokoll nicht direkt zu blockieren. Danach sendet der  $\text{NCAP}_C$  insgesamt neun Nachrichten per Broadcast an die STM, zunächst sechs hintereinander und nach einer Simulationszeit von 29 ms die restlichen drei. Für jede eingehende Nachricht senden  $\text{STM}_A$  und  $\text{STM}_S$  eine Antwort. Zusätzlich sendet der  $\text{STM}_A$  für jede empfangene Nachricht jeweils eine Statusnachricht. Die ersten 100 ms sind im Sequenzdiagramm in Abb. 5.5 dargestellt. Im gezeigten Beispiel wird die Kommunikation fehlerfrei abgearbeitet, Nachrichten vom Master werden unmittelbar beantwortet. Paketverluste für Nachrichtenpakete und Steuernachrichten werden nicht in Betracht gezogen.

Auf Ebene der Transportschicht werden die einzelnen Nachrichten in den drei verschiedenen Phasen des *MTS*-Protokolls aufgeteilt. Zusätzlich werden protokollbasierte Steuernachrichten versendet. Zu Beginn der Master- und Slave-Phasen wird ein Beacon versendet. In der Token-Phase erhalten alle Knoten, die in der Master-Phase eine Nachricht erhalten haben, ein Token, auf das der Master eine Antwort erwartet. Für die Status-Nachrichten vom Slave zum Master werden Acknowledgements verteilt, um den Erhalt der Nachrichten zu bestätigen. Die Abb. 5.6 zeigt den Verlauf der Kommunikation.

In diesem Beispiel werden acht der neun Anfragen in der Master-Phase an die STM übertragen. Das letzte zu sendende Paket wird jedoch nicht innerhalb der ersten Master-Phase gesendet. Der Datenaustausch wird nicht im ersten *MTS*-Zyklus abgeschlossen. In der Token-Phase werden 13 der erwarteten 16 Antworten zu den vorher gesendeten acht Anfragen eingesammelt. Hier wird die zeitliche Beschränkung durch die Konfiguration des Protokolls deutlich. In der Slave-Phase sendet  $\text{STM}_A$  für jede eingetroffene Nachricht eine Statusnachricht. Im unwahrscheinlichen Fall, dass sich mehr Statusnachrichten im Ausgangspuffer befinden als in der Zeit abgearbeitet werden können, wird die Sendung der Nachrichten vor dem Ende der Slave-Phase unterbrochen und die Master-Phase mit dem entsprechenden Beacon eingeleitet. In der zweiten Master-Phase wird im obigen Beispiel die bisher nicht gesendete neunte Nachricht an die STM gesendet. Danach werden dem  $\text{STM}_A$  die Acknowledgements für die erhaltenen Statusnachrichten vom  $\text{NCAP}_C$  zugestellt. Die übrigen erwarteten Antworten zu den in den beiden Master-Phasen versendeten Nachrichten werden in der zweiten Token-Phase eingesammelt. Der  $\text{STM}_A$  sendet für die zuletzt eingetroffene Nachricht eine weitere Statusnachricht und erhält dafür in der dritten Master-Phase das entsprechende Acknowledgement. Die Kommunikation ist damit im dritten Zyklus abgeschlossen.

Zur Optimierung lassen sich die konfigurierten Rechen- und Verzögerungszeiten betrachten. Dabei benötigen die Pakete, die in der Master- und der



**Abb. 5.6:** Das Resultat der Simulation verdeutlicht die Kommunikation gemäß dem *MTS*-Protokoll zwischen einem Master (NCAP<sub>C</sub>) und zwei Slave-Knoten (STM<sub>A</sub>, STM<sub>S</sub>). Der Datenaustausch wird durch das Versenden von acht Nachrichten in der ersten Master-Phase eingeleitet. Die Beantwortung der Broadcast-Nachricht erfolgt aufgeteilt auf die beiden darauffolgenden Token-Phasen. Für jede erhaltene Nachricht wird eine Statusnachricht durch den STM<sub>A</sub> an den NCAP<sub>C</sub> zurückgesendet. Eine weitere Nachricht sowie Acknowledgements werden durch den NCAP<sub>C</sub> in der zweiten Master-Phase übermittelt. Das letzte Acknowledgement wird in der dritten Master-Phase übertragen. Die Zykluszeiten des *MTS*-Protokolls sind äquidistant auf 100 ms aufgeteilt.

Slave-Phase gesendet werden, insgesamt 1,5 ms, bis ein neues Paket gesendet wird. In der Token-Phase ist der Abstand zwischen dem Senden zweier Token auf 2,5 ms festgelegt. Die Summe aus Rechen- und zusätzlicher Wartezeit innerhalb der Transportschicht beim Versenden der Daten wird hier berücksichtigt. Ohne diese zusätzliche Zeit würden die Pakete unmittelbar hintereinander versendet.

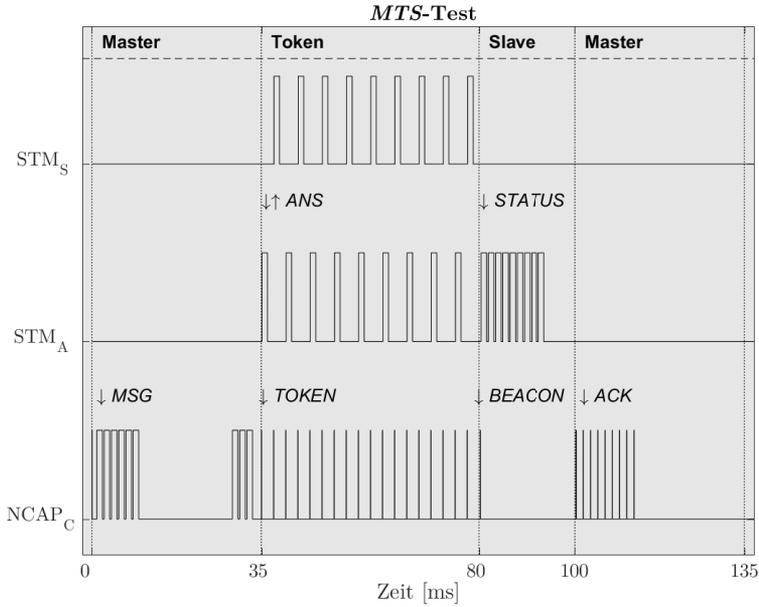
Die genäherten Zykluszeiten ergeben sich aus den in Tabelle 5.4 dargestellten Summen. Für das Versenden eines Beacons wird eine zusätzliche Zeit von einer Millisekunde angenommen. Die erste Master-Phase beinhaltet eine Wartezeit von 28 ms, in der sechs der neun Nachrichtenpakete versendet werden. Dies ist in der ersten Zeile berücksichtigt.

**Tabelle 5.4:** Aus den Sende- und Wartezeiten für die übertragenen Pakete in den drei verschiedenen *MTS*-Phasen lassen sich optimierte Zeitverteilungen für das vorgegebene Experiment ableiten.

Phase	Nachrichtentyp	Verteilung [ms]	Gesamt [ms]
Master	BEACON + MSG	$1,0 + 28 + 3 \cdot 1,5$	33,5
Token	TOKEN + ANS	$18 \cdot 1,5$	45,0
Slave	BEACON + STATUS	$1,0 + 9 \cdot 1,5$	14,5
Master	BEACON + ACK	$1,0 + 9 \cdot 1,5$	14,5

Nach Tabelle 5.4 ergibt sich für die Verteilung eines Zyklus auf 100 ms: 35 ms für die Master-Phase, 45 ms für die Token-Phase und die übrigen 20 ms für die Slave-Phase. Die Simulationsergebnisse in Abb. 5.7 zeigen, dass diese Konfiguration funktioniert. Die Kommunikation wird innerhalb von zwei Zyklen abgeschlossen.

Das Experiment belegt, dass nicht abgeschlossene Sendevorgänge im folgenden Kommunikationszyklus nachgeholt werden. Wird durch einen Fehler oder externe Einflüsse die Zeitbeschränkung der Kommunikationsphasen verletzt, so wird die Datenübertragung im nächstmöglichen Zyklus fortgeführt. Dabei werden die jeweiligen Daten nur in den dazu vorgesehenen Phasen gesendet. Eine unvorteilhafte Abstimmung der einzelnen Phasen auf die Applikation kann zu ungewollten Verzögerungen führen. Das *MTS*-Protokoll ist flexibel an die Gegebenheiten der Anwendung anpassbar um eine echtzeitige Übertragung zu gewährleisten. Für die konkrete Anwendung ist jedoch eine spezifische



**Abb. 5.7:** In diesem Simulationsbeispiel ist die Kommunikation innerhalb von zwei Zyklen zwischen einem NCAP<sub>C</sub> und je einem STM<sub>A</sub> und STM<sub>S</sub> abgeschlossen. Der Datenaustausch wird durch das Versenden von neun Nachrichten in der ersten Master-Phase eingeleitet. Die Beantwortung der Broadcast-Nachricht erfolgt durch STM<sub>A</sub> und STM<sub>S</sub> in der Token-Phase. Für jede erhaltene Nachricht wird eine Statusnachricht durch den STM<sub>A</sub> an den NCAP<sub>C</sub> zurückgesendet. Der Erhalt dieser wird durch Acknowledgements in der folgenden Master-Phase bestätigt. Die Zykluszeiten des MTS-Protokolls sind auf die Datenübertragung abgestimmt.

Abstimmung notwendig.

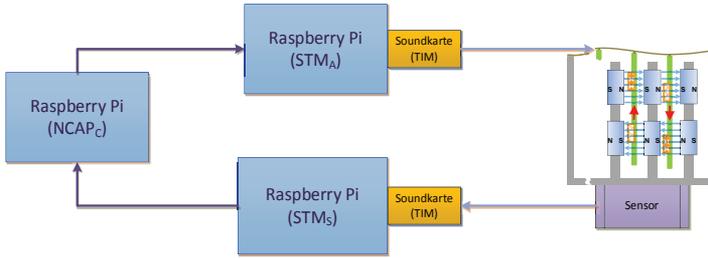
Die reine Betrachtung des *MTS*-Protokolls unter Zuhilfenahme von Übertragungszeiten der MAC-Ebene für die verschiedenen Pakete zeigt die Flexibilität des Protokolls. Die Länge der einzelnen Kommunikationsphasen (Master, Token, Slave) sind adaptierbar. Kollisionen können je nach verwendetem Protokoll auf der Netzwerkschicht in der Slave-Phase auftreten. In einem switched Ethernet basierten Netzwerk werden diese vermieden, jedoch ist die Anzahl der Nachrichten limitiert durch die Puffergröße im Switch. Das flexible, echtzeitfähige Transportprotokoll wird für die Untersuchung von Modellen in Simulationen für die Entwicklung echtzeitfähiger Aktuator- und Sensornetzwerke in kaskadierten Regelkreisen eingesetzt. Für die Verwendung im Regelkreis wird in den folgenden Kapiteln für die Transportschicht das geschachtelte *MST-MTS*-Protokoll verwendet. Die grundsätzliche Funktionsweise ändert sich dabei nicht. Das  $\text{NCAP}_I$ - $\text{NCAP}_C$ -Netzwerk mit der *MST*-Konfiguration ermöglicht die Beantwortung der übertragenen Daten der  $\text{NCAP}_C$  an die STM, sodass Anfrage und Antwort innerhalb eines Zyklus alle Knoten passieren und der externen Regelung bereitgestellt werden.

### 5.2.3 Validierung der Anwendungsschicht

Für die Untersuchung des IEEE 1451 Protokolls wurde die Beispielanwendung des *National Institute of Standards and Technology* (NIST) in der Raspberry Pi Umgebung implementiert. Die Open1451 Gaithersburg [90] Implementierung ist eine Umsetzung des IEEE 1451.1 Substandards und stellt mit einigen Modifikationen eine Hardware-Repräsentation für die Verwendung im inneren Regelkreis dar. Die Ansteuerung der Aktuatoren, Auslese der Sensoren und die Kommunikationsbasis sind dabei in der Software integriert. Innerhalb des Protokollstapels wird unterhalb des IEEE 1451.1 ein TCP/IP basiertes Protokoll verwendet. Die Umsetzung der analogen Signale erfolgt unter Verwendung der Soundkarte *Creative Sound Blaster Play* [22] im Verbund mit den Treibern der *Advanced Linux Sound Architecture* [6].

Das Ziel der Untersuchung ist die Parameteridentifikation für die Modellierung in TrueTime. Dabei wird die Übertragungszeit durch die IEEE 1451.1 Anwendungsschicht gemessen und die Verwendung für die Ansteuerung der Aktuatoren und Auslese der Sensoren erprobt.

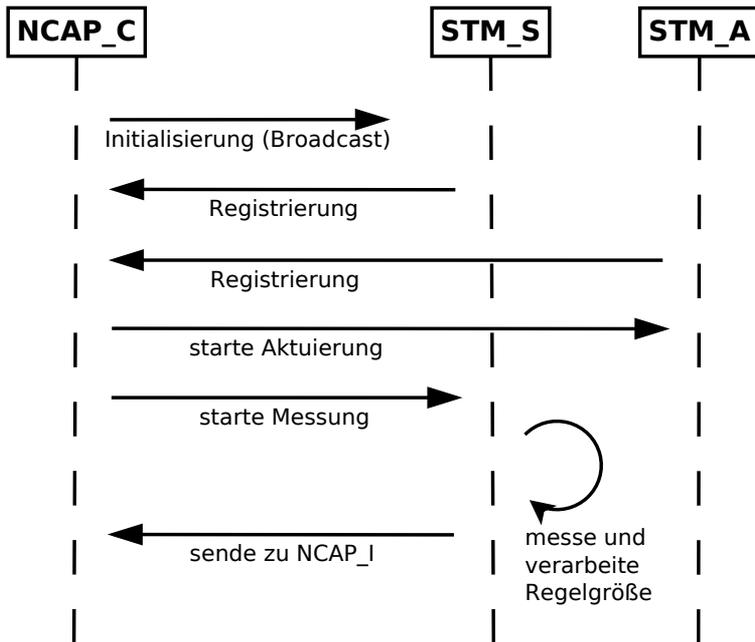
In diesem Fall besteht die Testumgebung aus drei verschiedenen Knoten. Entsprechend des Modells sind das ein  $\text{NCAP}_C$ , ein  $\text{STM}_A$  mit analogem Ausgang und ein  $\text{STM}_S$  mit analogem Eingang. Der  $\text{NCAP}_C$  kontrolliert dabei die zwei



**Abb. 5.8:** Schematische Darstellung der Vernetzung der Raspberry Pi-Knoten für die Verwendung als IEEE 1451.1 Testumgebung. Der  $STM_A$  wird durch den  $NCAP_C$  aufgefordert, analoge Signale zur Steuerung des Aktuator-Systems zu generieren, während der  $STM_S$  Sensordaten aufzeichnet, verarbeitet und nach Aufforderung an den  $NCAP_C$  übergibt.

STM-Knoten, die mit einer Soundkarte verbunden sind (siehe Abb. 5.8). Daten für die Steuerung sind im vorgefertigten Paketrahmen des IEEE 1451.1 Protokolls integriert. Die in diesem Protokoll eingebetteten Kommunikationsabläufe Publish-Subscribe sind über Multicasts realisiert, für die Verwendung des Client-Server-basierten Datenaustausches werden die Nachrichten direkt an die Knoten adressiert. Die Kommunikation erfolgt dabei mithilfe von TCP/IP-Paketen auf Basis des *Adaptive Communication Environments* (ACE) [76]. Das Kommunikationsschema ist in Abb. 5.9 dargestellt. Zuerst meldet der  $NCAP_C$ , dass er auf die Anmeldung der STM-Knoten wartet. Diese registrieren sich beim  $NCAP_C$ . Daraufhin werden die Aufforderungen zur Aktuierung und zur Sensordatenerfassung versendet. Die erfassten Daten werden anschließend regelmäßig nach der Vorverarbeitung an den  $NCAP_C$  übermittelt.

Die Kommunikation der Prozesse innerhalb der Knoten ist entsprechend der Vorgaben aus dem Protokollstandard zwischen `FBlock`, `TBlock`, `NCAPBlock` implementiert. Die Übertragungszeit der Pakete durch die IEEE 1451.1 Ebene wird auf der Seite der STM sowohl beim Eintreffen als auch beim Versenden von Nachrichten gemessen. Es werden dabei für jede Messung zwei Messpunkte betrachtet, zum einen vor der Übergabe eines Paketes an das IEEE 1451.1 Protokoll und zum anderen nach der Weitergabe des Paketes an die darunter liegende Kommunikationsschicht. Die Übertragungszeit durch die Anwendungsschicht von der Übergabe der Parameter bis zum Versenden wird daraus abgeleitet. Die Ergebnisse der Messung sind in Tabelle 5.5 dar-



**Abb. 5.9:** Kommunikationsablauf zwischen NCAP<sub>C</sub>, STM<sub>A</sub> und STM<sub>S</sub> unter Verwendung des IEEE 1451.1 Anwendungsprotokolls.

gestellt, sowohl beim Anmeldevorgang, als auch bei der Datenübertragung wird die Übertragungszeit durch das IEEE 1451.1 Protokoll gemessen und die Werte gemittelt.

**Tabelle 5.5:** Evaluierung der Übertragungsparameter durch die IEEE 1451.1-basierte Anwendungsschicht. Die Übertragungszeit wurde aus 50 Messungen sowohl zum Anmeldevorgang des  $STM_A$  als auch zur Übertragung der Daten der  $STM_S$  bestimmt.

Parameter	Wert
Mittelwert Anmeldevorgang	2,5 ms
Standardabweichung Anmeldevorgang	0,5 ms
Mittelwert Datenübertragung	3,5 ms
Standardabweichung Datenübertragung	2,1 ms

Die gemessenen Werte zeigen, dass drei Millisekunden eine gute Näherung für die Übertragungszeit durch das IEEE 1451.1 Protokoll ist. Bei der Messung der Übertragungszeit im Anmeldevorgang wurde nur eine geringe Variation von 0,5 ms festgestellt. Für die Zeiten der Datenübertragung ergibt sich eine hohe Standardabweichung, was allerdings nur auf eine schlechte Abstimmung zwischen Kommunikationsprotokoll und Sensorabtastung hindeutet.

### 5.3 Validierung des Netzwerkmodells

Das Ziel der Modellvalidierung ist es, das vorhergesagte Verhalten innerhalb der Simulation nachzuweisen. Aufgrund der frei wählbaren Teilnehmeranzahl und -verteilung im Netzwerk lässt sich die Funktion des Netzwerkes nicht allgemein nachweisen.

Für die gegebene Applikation wird ein Netzwerkmodell mithilfe des *CreNeMo*-Werkzeugs erstellt und anschließend die Simulation durchgeführt. Als externe Anforderung wird ein Regler gewählt, der auf die Kommunikation abgestimmt ist, die Stellgrößen in jedem Zyklus aktualisiert und Messdaten in jedem Zyklus entgegennimmt.

### 5.3.1 Evaluierung konkreter Netzwerkkonfigurationen

Die im Folgenden erzeugten Netzwerke entsprechen dem entwickelten Kommunikationsmodell aus switched Ethernet in einer Baumstruktur, bestehend aus einem einzelnen  $\text{NCAP}_I$ , mehreren  $\text{NCAP}_C$ ,  $\text{STM}_A$  und  $\text{STM}_S$ . Die *MST-MTS*-Protokollkonfiguration wird auf der Transportschicht eingesetzt und die Modellparameter aus dem IEEE 1451.1 Protokoll werden auf der Anwendungsschicht miteinbezogen. Die Paketgröße auf der Netzwerkschicht entspricht 1500 Bytes für Datenpakete und 64 Bytes für protokollbasierte Nachrichten. Die Datenübertragung erfolgt wie im Sequenzdiagramm in Abb. 3.10 dargestellt. Zu Beginn findet zunächst der Registrierungsprozess statt, bei dem sich die Slave-Knoten bei ihrem Master anmelden. Danach wird auf eingehende Daten der externen Strömungsregelung gewartet. Die bisher validierten Modellparameter werden zur Generierung des konkreten Modells in das *CreNeMo*-Werkzeug eingetragen. Sie sind in Tabelle 5.6 aufgeführt.

**Tabelle 5.6:** Validierte Parameter zur Verwendung im Netzwerk.

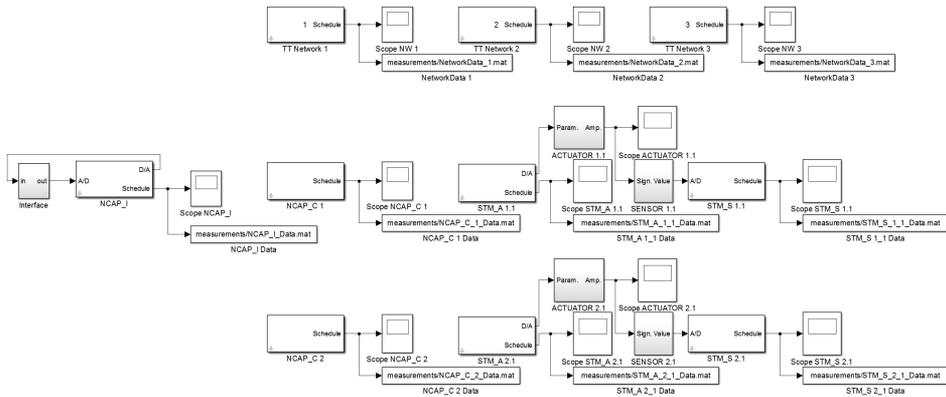
Parameter	Wert
Netzwerktyp	switched Ethernet
Netzwerkorganisation	zwei Ebenen (Baumstruktur)
Interface Subsysteme	$\text{NCAP}_I$ , $\text{STM}_A$ , $\text{STM}_S$
Asynchronität	Nein
Datenrate	6,6 MBits/s
Minimale Paketgröße	64 Byte
Paketverlustrate	0%
Switchpuffergröße	256 KB
Switchverhalten	Verwerfen bei Überlauf
IEEE 1451.1 Übertragungszeit	3 ms

Es wird das Beispielnetzwerk mit den Parametern aus Tabelle 5.6 evaluiert. Die Zahlenwerte entsprechen den bisher validierten Parametern für die Netzwerk- und Anwendungsschicht und werden in der Simulation nicht verändert. Die Paketverlustrate wird auf null Prozent gesetzt und die Knoten sind von vorneherein gleich getaktet, sodass eine Asynchronität verhindert wird. Die Einflüsse von Fehlern in der Kommunikation und die Möglichkeiten der Synchronisation werden nachfolgend gesondert betrachtet.

**Tabelle 5.7:** Anzahl der Knoten im Netzwerk, die in einer Baumstruktur zusammengefasst werden.

Knoten	Anzahl
$NCAP_I$	1
$NCAP_C$	2
$STM_A$ pro $NCAP_C$	1
$STM_S$ pro $NCAP_C$	1

Die für das konkrete Modell frei zu wählenden Parameter sind die Knotenanzahl sowie deren Verteilung auf  $NCAP_I$ ,  $NCAP_C$ ,  $STM_A$  und  $STM_S$  und die Zykluszeiten innerhalb des *MST-MTS*-Protokolls. Als Beispiel wird ein kleines Netzwerk mit einem Schnittstellenknoten, zwei  $NCAP_C$  und jeweils einem  $STM_A$  und einem  $STM_S$  (siehe Tabelle 5.7) generiert. Es wird somit ein Modell mit sieben Knoten, verteilt auf drei Netzwerke, betrachtet. Das entsprechend generierte Simulinkmodell ist in Abb. 5.10 dargestellt.



**Abb. 5.10:** Das TrueTime-Modell zeigt sieben Kernel-Blöcke und drei Netzwerk-Blöcke. Ein  $NCAP_I$  und zwei  $NCAP_C$  mit je einem  $STM_A$  und einem  $STM_S$ .

Zunächst wird der  $NCAP_I$ -Schnittstelle die Abtastrate der  $STM_S$ -Messung mitgeteilt. Diese soll in jedem Zyklus unabhängig von der Zykluslänge ge-

nau einmal abgearbeitet werden. Eine Anforderung der Applikation ist die entsprechende Reihenfolge, sodass im ersten Zyklus Messdaten bereitgestellt werden. In jedem weiteren Zyklus werden zuerst die Aktuierungs-Parameter gestellt und danach die Messung durchgeführt. Die von der externen Regelung eingehenden Stellgrößen für die Aktuierung werden bezüglich einmalig festgelegter Limitierungen im  $\text{NCAP}_I$  geprüft und nur bei positivem Prüfungsergebnis an den zuständigen  $\text{NCAP}_C$ -Bereich verteilt. Bei festgestellter Verletzung der Limitierung wird dies der externen Regelung mitgeteilt und die Aktuierung nicht verändert (siehe Kapitel 5.3.4). Die Aufforderung zur Messung wird nach Vorgabe der Abtastrate durch den  $\text{NCAP}_I$  im Netzwerk initiiert. Die umgesetzten Parameter und zugehörigen Messdaten nach Stellen der Aktuierungs-Parameter nimmt die externe Regelung in einem Paket entgegen. Bei der Übertragung und Umsetzung der Daten an den STM ist die Simulation an dieser Stelle abgesehen von der Übertragungszeit auf der Anwendungsschicht nicht durch zusätzliche Rechenzeit verzögert.

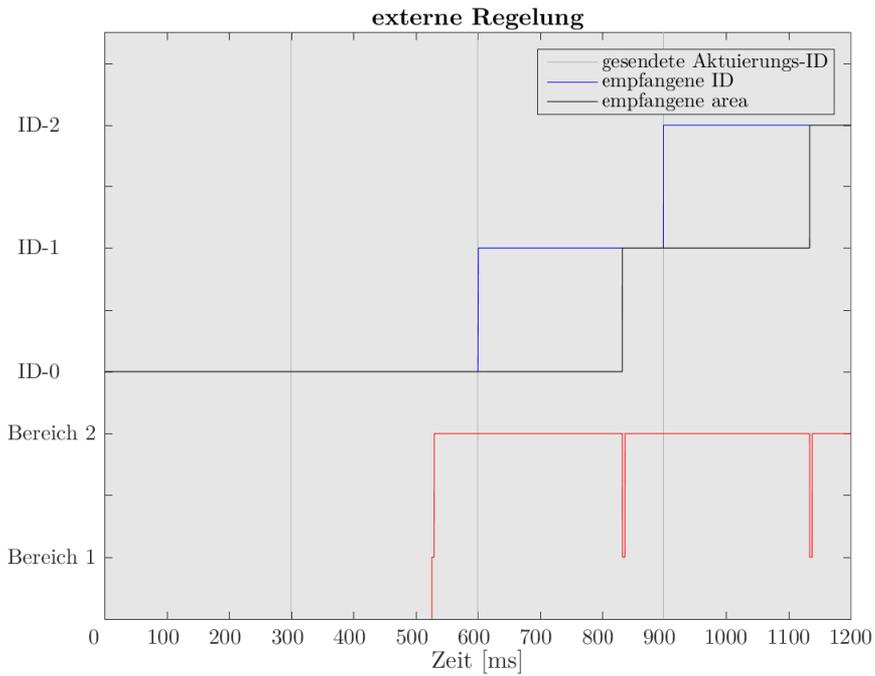
**Tabelle 5.8:** Konservative *MTS*-Zeitverteilung für ein reduziertes Netzwerk.

Zykluszeit [ms]	$M_1-S_1-T_1$ [ms]	$M_2-T_2-S_2$ [ms]
300	100 – 125 – 75	100 – 75 – 125

Die *MST-MTS*-Konfiguration muss mit der Applikation abgestimmt werden, um die erwartete Funktionsweise nachzuweisen. Die experimentelle Bestimmung der Zeiten unter Beachtung der Randbedingungen bietet eine Möglichkeit zur entsprechenden Konfiguration eines Netzwerkes. Dabei muss beispielsweise die Überdeckung der beiden Slave-Phasen (siehe Kapitel 3.4.3) berücksichtigt werden.

Eine konservative Zeitverteilung für die Transportschicht ist in Tabelle 5.8 dargestellt. Insgesamt ist der Zyklus von 300 ms wie folgt aufgeteilt: 100 ms Master-Phasen, 75 ms Token-Phasen ( $T_1$  und  $T_2$ ) und 125 ms die beiden Slave-Phasen ( $S_1$  und  $S_2$ ). Diese Konfiguration erzeugt die notwendige Überdeckung, sodass der Registrierungsprozess im ersten Zyklus vollständig abgeschlossen wird. Nach der Festlegung der Zeitparameter für die Transportschicht wird das Netzwerk in Kombination mit einer angekoppelten externen Reglerkomponente auf seine Funktionsweise geprüft.

Um die korrekte Arbeitsweise nachzuweisen, werden 50 Zyklen (15 s) simuliert. Zur Prüfung der korrekten Übertragung werden die an der Schnittstelle



**Abb. 5.11:** Der Datenaustausch zwischen externer Regelung und NCAP<sub>1</sub> wird durch die IDs der Nachrichten und den `area`-Parameter beschrieben. Die *MTS*-Zykluszeit beträgt 300 ms. Die ersten vier Zyklen sind dargestellt. Die Versendung eines Paketes von der externen Regelung wird mit Paketen mit der gleichen ID beantwortet. Dabei wird für jeden Bereich ein Paket erhalten.

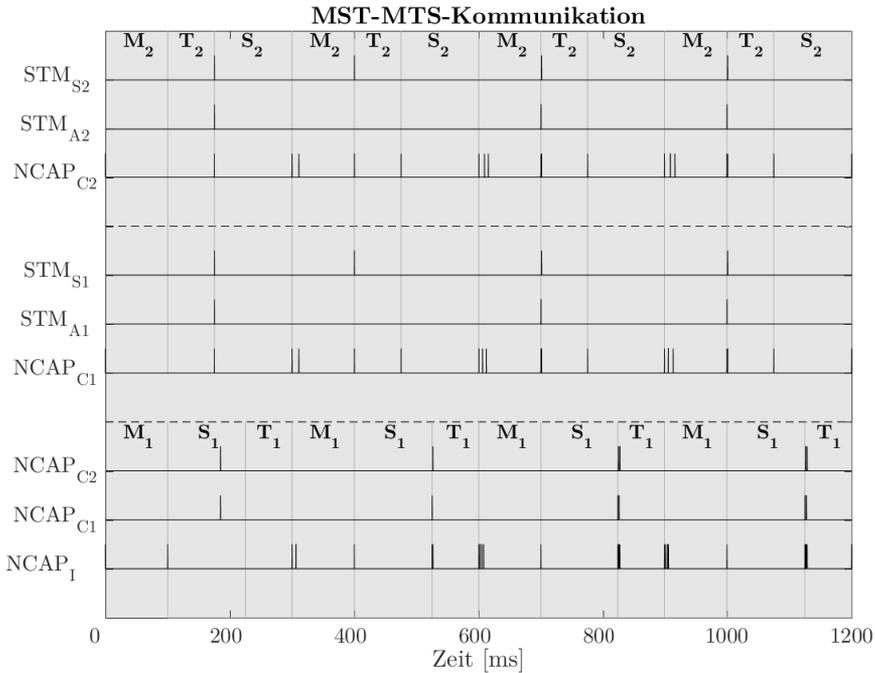
der externen Regelung aus- und eingehenden Daten analysiert. Es werden sowohl die ID als auch der **area**-Parameter betrachtet (siehe Kapitel 3.1.2). Die Abb. 5.11 zeigt die Veränderung dieser Parameter aus der Sicht der externen Regelung innerhalb der ersten vier Zyklen. Die Übertragung der Messdaten erfolgt mit ID-0 und wird nach etwas mehr als 500 ms beantwortet. Dies kann durch die Veränderung des **area**-Parameters beobachtet werden. Im Zeitpunkt 600 ms werden mit der ID-1 die Aktuierungs-Parameter geändert. Die gemessene Zeit zwischen dem Versenden der ersten Nachricht einer ID bis zur ersten Antwort beträgt 233 ms. Der vollständige Erhalt ist nach weiteren vier Millisekunden sichergestellt. Die Gesamtzeit zur Nachrichtenbeantwortung beträgt 237 ms und liegt innerhalb der Zykluszeit von 300 ms. Dieser Vorgang wird in einem Zyklus von 300 ms erneut angestoßen und rechtzeitig beantwortet.

Abb. 5.12 zeigt vier Kommunikationszyklen. Im ersten Zyklus melden sich die einzelnen Slaves nach dem Beginn der Slave-Phase bei ihrem Master an, zuerst die STM bei ihren  $NCAP_C$  und danach die  $NCAP_C$  beim  $NCAP_I$ . Im zweiten Zyklus wird die Aufforderung zur Messdatenerfassung an die Sensor-Knoten gesendet. In den weiteren Zyklen werden zuerst die Aktuator-Knoten mit Parametern bedient, bevor die Sensor-Knoten eine Nachricht erhalten und ihre Messdaten eingeholt werden. Die Netzwerkauslastung liegt bei 1,41% im  $NCAP_I$ - $NCAP_C$ -Netzwerk und jeweils 0,98% in den  $NCAP_C$ -STM-Netzwerken.

Die Zusammenführung der einzelnen Elemente und die Validierung des generierten Netzwerkmodells in einer Simulation zeigen, dass die Komponenten im Netzwerk gut aufeinander abgestimmt sind und das Modell noch auf Anforderungen der externen Regelung abgestimmt werden kann. Die Netzwerkgenerierung und -variation der Modellparameter bietet breite Möglichkeiten zur Konfiguration und schafft die Voraussetzung zur Erzeugung von echtzeitfähigen Aktuator- und Sensornetzwerkmodellen. Anhand der wenigen Netzwerkeignisse und der geringen Netzwerkauslastung ist festzustellen, dass die Zeitverteilung Möglichkeiten zur Optimierung bietet.

Allgemein gilt in einem Ethernet-Netzwerk für die Netzwerkauslastung ein durchschnittlicher Wert von 60% als kritische Marke. Jedoch ist zu beachten, dass im vorliegenden Netzwerk zu Beginn beim Registrierungsprozess eine kurzzeitige Auslastung von 100% erreicht wird. Eine durchschnittliche Auslastung in diesem Netzwerk von bis zu zehn Prozent wird als unkritisch angesehen. Allgemein gilt je höher die Netzwerkauslastung ist, desto effizienter wird das Medium verwendet.

Die Optimierung kann sowohl durch zusätzliche Netzwerkteilnehmer als auch durch die Verbesserung der Zykluszeiten erfolgen. In beiden Fällen ist es mög-



**Abb. 5.12:** Der Datenaustausch innerhalb des Netzwerkes zwischen dem NCAP<sub>I</sub> und den NCAP<sub>C</sub> sowie in den beiden NCAP<sub>C</sub>-STM-Netzwerken wird durch den TrueTime-Graphen visualisiert. Die *MTS*-Zykluszeit beträgt 300 ms. Die ersten vier Zyklen sind dargestellt. Im ersten Zyklus erfolgt die Registrierung der STM an die jeweiligen NCAP<sub>C</sub> und danach die Registrierung der NCAP<sub>C</sub> an den NCAP<sub>I</sub>. Die Datenverteilung und -zusammenführung folgt dem *MTS*-Protokoll. Die von der externen Regelung vorgegebenen Werte werden verteilt und die Antwortpakete sowie die Messdaten der STM<sub>S</sub> werden innerhalb eines Zyklus im NCAP<sub>I</sub> empfangen.

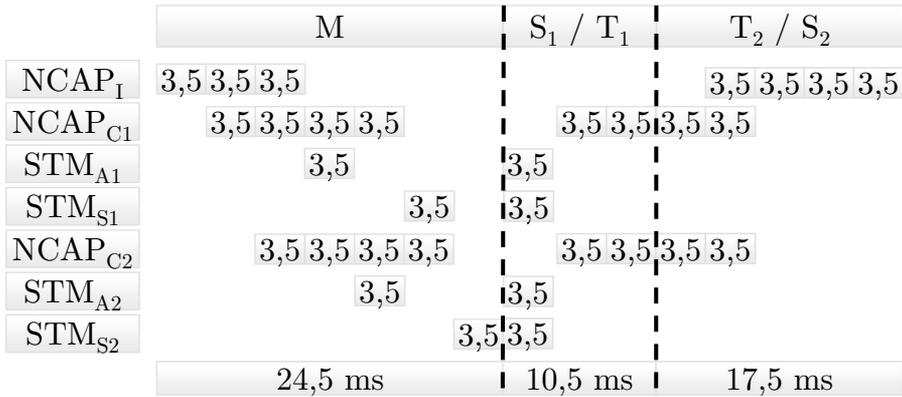
lich, weiterhin eine funktionierende Simulation aufrechtzuerhalten, die den gewünschten Anforderungen entspricht.

### 5.3.2 Analytischer Ansatz zur Optimierung der Zykluszeiten

Unter Berücksichtigung vereinfachter Annahmen lässt sich die Zeitverteilung für die *MST-MTS*-Konfiguration im beschriebenen Netzwerk verbessern. Das zu untersuchende Netzwerk besteht aus einem  $\text{NCAP}_I$  als Master zweier  $\text{NCAP}_C$  und jeweils einem  $\text{STM}_A$  und  $\text{STM}_S$ -Knoten als Slaves der  $\text{NCAP}_C$ -Knoten. Es wird eine Berechnungszeit und Übertragungszeit auf der Anwendungsschicht sowie eine zusätzliche Zeitspanne für Transport- und Netzwerkschicht berücksichtigt. Die Übertragungszeit für Pakete durch die IEEE 1451.1 Schicht wird mit drei Millisekunden pro Paket jeweils für Senden und Empfangen angenommen. Zusätzliche Rechen- und Bearbeitungszeit auf der Anwendungs-, Transport- und Netzwerkschicht werden mit weniger als 0,5 ms pro Paket berücksichtigt. Mithilfe dieser Annahmen werden Zykluszeiten für die einzelnen Übertragungsphasen  $M_1-S_1-T_1$  beziehungsweise  $M_2-T_2-S_2$  berechnet. Um einen reibungslosen Ablauf zu gewährleisten, wird die benötigte Zeit für den zu Beginn der Simulation stattfindenden, automatisierten Registrierungsprozess mit in die Berechnung eingegliedert, sodass die erhaltenen Zykluszeiten sofort in eine Simulation übertragbar sind und die Funktionsweise validiert werden kann.

Im vorliegenden Beispiel wird von einer Inhomogenität ausgegangen, sodass jeder  $\text{NCAP}_C$  als eigener Bereich spezifische Parameter erhält. Abb. 5.13 zeigt eine Rechen- und Übertragungszeitverteilung für den Informationsaustausch während eines typischen Kommunikationszyklus. Diese basiert auf den Sende- und Verarbeitungszeiten der verschiedenen Kommunikationsschichten, die mit 3,5 ms pro Datenpaket abgeschätzt werden.

Zu Beginn sendet  $\text{NCAP}_I$  drei Pakete zu jedem seiner Slaves. Die  $\text{NCAP}_C$  erhalten ausgehend vom  $\text{NCAP}_I$  in der Master-Phase zuerst jeweils ein Paket für die Verteilung der Aktuierungs-Parameter in ihrem Bereich. Die Aufforderung zur Aufnahme der Sensordaten erfolgt automatisiert und einheitlich für alle Bereiche als Broadcast, sobald die entsprechende Messzykluszeit im  $\text{NCAP}_I$  erreicht ist. Die Pakete werden im  $\text{NCAP}_C$  angenommen und entsprechend ihres Bestimmungsortes an  $\text{STM}_A$  und  $\text{STM}_S$  weitergesendet. Der Empfang der Pakete findet noch innerhalb der Master-Phase statt.



**Abb. 5.13:** Die Ableitung der Zykluszeiten aus den Sendevorgängen basiert auf der Annahme, dass die Übertragungs- und Verarbeitungszeit jedes Pakets mit 3,5 s als obere Schranke abgeschätzt werden kann. Die Übertragung und Verarbeitung wird als parallel angenommen. Das Resultat ist eine Zeitverteilung für die *MST-MTS*-Phasen in der Transportschicht für einen Kommunikationszyklus vom NCAP<sub>1</sub> zu den STM und zurück.

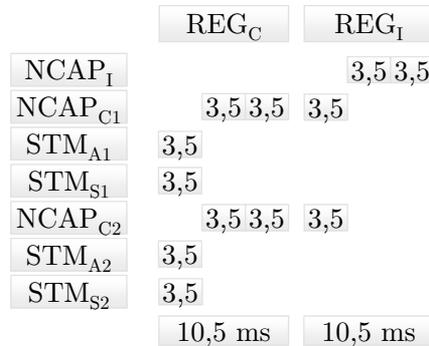
Die nächste Phase ist aufgeteilt in  $S_1$  für NCAP<sub>1</sub> und NCAP<sub>C</sub>, und  $T_1$  für NCAP<sub>C</sub> und die STM-Knoten. Entsprechend der Bestimmung der Knoten und der Inhalte der Datenpakete werden die bereitgestellten Pakete im STM an den NCAP<sub>C</sub> zurückgesendet. Es wird also pro STM ein Paket versendet und somit im vorgegebenen Beispiel zwei Pakete im NCAP<sub>C</sub> angenommen.

Die dritte Phase ist dann in  $T_2$  und  $S_2$  unterteilt. Während in dieser Berechnung keine zusätzlichen Aktivitäten in den Slave-Phasen berücksichtigt werden, senden die NCAP<sub>C</sub> Knoten nach der Aufforderung durch den NCAP<sub>1</sub> jeweils ein Paket für die Aktuierung und ein Paket für die Sensordaten zurück.

Die minimalen Zykluszeiten der einzelnen Kommunikationsphasen lassen sich mit diesem Ansatz wie folgt bestimmen:

- $M = 24,5 \text{ ms}$
- $S_1, T_1 = 10,5 \text{ ms}$
- $T_2, S_2 = 17,5 \text{ ms}$

Das Senden der Slave-Knoten in den Token-Phasen erfolgt nicht simultan, jedoch mit so geringem Zeitversatz, dass dies als simultan beschrieben werden kann. Für eine große Anzahl Netzwerkteilnehmer ist dies allerdings zu beachten.



**Abb. 5.14:** Die Registrierung der STM beim NCAP<sub>C</sub> sowie die der NCAP<sub>C</sub> beim NCAP<sub>I</sub> wird gesondert betrachtet. Die Übertragung der Pakete wird als parallel angenommen. Das Ergebnis ist die Zeit, die die Knoten für den Registrierungsprozess mindestens benötigen.

Für den Registrierungsprozess, der in Abb. 5.14 dargestellt ist, müssen sich  $S_1$  und  $S_2$  überlagern. Um diese Überlagerung zu erreichen, wird die Anmeldezeit von 10,5 ms jeweils zu den Phasenlängen  $S_1$  und  $S_2$  hinzuaddiert. Zur zeitlichen Abstimmung beider Zyklen werden diese also auf eine Länge von 61 ms erweitert (siehe Tabelle 5.9). Dabei werden die herausgearbeiteten Zeiten auf volle Millisekunden aufgerundet, bevor die Zusammenrechnung erfolgt.

Die Simulation des modifizierten Modells mit den so bestimmten Zykluszeiten von 65 ms, simuliert in 50 Zyklen (3,25 s) verhält sich analog zum vorherigen Modell mit der Zykluszeit von 300 ms. Jedoch wird für das optimierte Modell ein anderer Regler mit einer Zykluszeit von 65 ms verwendet. Abb. 5.15a zeigt die Nachrichtenübertragung innerhalb der ersten vier Kommunikationszyklen im Detail. Die Pakete werden hintereinander durch das Netzwerk geleitet. Gesendete und empfangene Pakete werden innerhalb eines Kommunikationszyklus verarbeitet und die Pufferspeicher sind am Ende der einzelnen Kommunikationsphasen und des Kommunikationszyklus geleert (siehe Abb. 5.15). Die vier Zyklen sind nach 260 ms abgeschlossen. Die Netzwerkauslastung in

**Tabelle 5.9:** Die Bestimmung der Zykluszeiten für die *MST-MTS*-Kommunikation basiert aus der Berechnung des Kommunikationszyklus und der Berücksichtigung des Registrierungsprozesses.

	$M$	$S_1$	$T_1$	$S_2$	$T_2$	$\Sigma(\text{Zyklus})$
min. Zeit [ms]	25	11	11	18	18	54
Reg. Zeit [ms]		11		11	11	
Gesamtzeit [ms]	25	22	11	18	29	65

der Simulation beträgt 5,85% für das NCAP<sub>I</sub>-NCAP<sub>C</sub>-Netzwerk und 4,34% jeweils für die NCAP<sub>C</sub>-STM-Netzwerke.

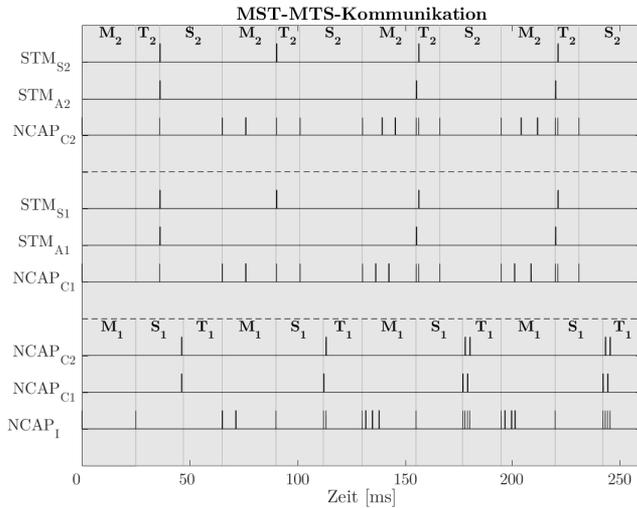
Die gezeigte analytische Berechnung der Zykluszeiten liefert eine Näherung für ein funktionierendes Netzwerk und verringert die vorher experimentell bestimmte, konservative Zeitverteilung auf etwa ein Fünftel. Die berechneten Parameter werden am Modell in einer Simulation geprüft. Zusätzliche Randbedingungen wie mögliche Paketverluste, Verzögerungen oder Fehler sind hier zunächst nicht berücksichtigt.

### 5.3.3 Synchronisation

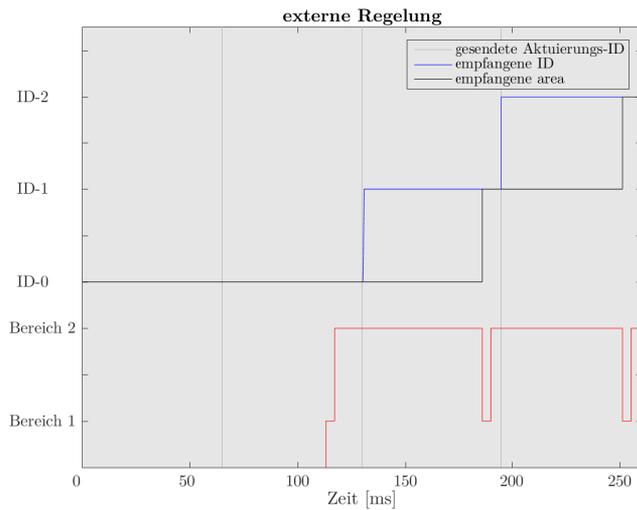
Die zeitliche Synchronisation aller Netzwerkknoten wird mittels der oben beschriebenen Modellkonfiguration evaluiert. Dabei muss in der TrueTime-Simulation sowohl der zeitliche Versatz zwischen den Uhren beim Start als auch eine grundsätzliche Abweichung der Uhren von der Simulationsgeschwindigkeit berücksichtigt werden. Die Vorgabe des zeitlichen Taktes obliegt dem NCAP<sub>I</sub>, dieser Knoten wird also von den Synchronisationsmechanismen nicht beeinflusst.

Für die Synchronisation sind unterschiedliche Genauigkeiten zum einen für die Model-in-the-Loop-Simulation und zum anderen für die Applikation des Netzwerkes in der Strömungsregelung gefordert. Es muss sichergestellt werden, dass eine übliche Genauigkeit beispielsweise der Raspberry Pi Plattform von 100 ppm [78] nicht die Netzwerkkommunikation beeinflusst.

Für die Anwendung in der Strömungsregelung ist eine Abweichung der Oberflächenwelle von der vorgegebenen Trajektorie von maximal fünf Prozent vor-



(a) Darstellung der Kommunikation aufgeteilt auf drei verschiedene Netzwerke.



(b) Darstellung der Kommunikation zwischen externer Regelung und NCAP<sub>I</sub>.

**Abb. 5.15:** Die optimierte *MTS*-Zykluszeit beträgt 65 ms, es sind jeweils die ersten vier Zyklen dargestellt.

gegeben (siehe Kapitel 4). Um die Wellenregelung nicht zu beeinflussen, wird eine Abweichung der Phase durch die zeitliche Asynchronität des Knotens von maximal 0,75% vorgegeben. Bei einer Aktuierungsfrequenz von 81 Hz ist das eine Abweichung von 0,3375 Grad. Diese kann durch eine Synchronisation mit der Genauigkeit von  $15\mu\text{s}$  analog zu Gleichung 5.6 erreicht werden.

$$\Delta t = \frac{\phi}{f \cdot 2\pi} \quad (5.6)$$

In der TrueTime-Umgebung können sowohl positive als auch negative Abweichungen simuliert werden. Der zeitliche Versatz wird durch einen Absoluten Wert angegeben. Dieser beschreibt, ob die Startzeit des Knotens später (positiv) oder früher in Relation zur Simulationszeit gewählt wird. Die relative Abweichung der Uhr von der Simulationsgeschwindigkeit wird prozentual angegeben.

Positive zeitliche Unterschiede der Slave- von den Master-Knoten können mit den in der Simulation verwendeten Methoden nicht ausgeglichen werden. Da die Kommunikationssteuerung vom Master ausgeht und mit der Synchronisation einhergeht, ist eine negative absolute Abweichung von der Simulationszeit jedoch kompensierbar.

Die maximale Abweichung bei 100 ppm beträgt  $10^{-5}$ s in einer Millisekunde. Sowohl für die Netzwerkkommunikation als auch für die Anwendung in der Strömungsregelung genügt die vorgestellte Synchronisation, sofern sie in einem regelmäßigen Zyklus von weniger als 100 ms erfolgt.

### 5.3.4 Fehlerbehandlung

Zur Evaluation der Fehlerbehandlung wird ein reduziertes Netzwerkmodell, bestehend aus einem  $\text{NCAP}_I$ , einem  $\text{NCAP}_C$  und jeweils einem  $\text{STM}_A$  und  $\text{STM}_S$  betrachtet, Die Netzwerkteilnehmer sind entsprechend der Baumstruktur (siehe Abb. 3.6) auf zwei verschiedene Netzwerke verteilt. Die entwickelten und im Modell implementierten Strategien zur Fehlerbehandlung werden im Folgenden nacheinander betrachtet und evaluiert.

### Plausibilitätsprüfung der eingehenden Parameter

Bei Über- oder Unterschreitung der Grenzen für die von der externen Regelung übertragenen Stellgrößen der Aktuatoren werden die erhaltenen Parameter nicht an die  $NCAP_C$  und  $STM_A$  weitergegeben. Stattdessen werden Fehlercodes gesetzt und in einer direkten Antwort an die externe Regelung zurückgegeben. In der Simulation verzögert die Fehlerbehandlung die Anwendung neuer Aktuierungs-Parameter um bis zu einen Zyklus, sofern die Stellgrößen innerhalb des Zyklus durch die externe Regelung korrigiert werden.

### Strategie zum Umgang mit Paketverlusten bei der Nachrichtenübermittlung

In der oben beschriebenen reduzierten Anwendung ohne die Möglichkeit von Paketverlusten wird zuerst die Registrierung der Slaves an die Master-Knoten durchgeführt. Danach wird ein Paket mit Aktuierungs-Parametern und ein Paket zur Messaufforderung an die jeweiligen STM gesendet. Die Antworten werden eingesammelt und die Kommunikation ist nach zwei Zyklen abgeschlossen. Die Verteilung der  $MST$ - $MTS$ -Zeiten ist in Tabelle 5.10 aufgeführt. Die Zeit eines einzelnen Zyklus beträgt 160 ms.

**Tabelle 5.10:**  $MST$ - $MTS$ -Zeitverteilung für die Simulation zur Validierung der Fehlerbehandlung.

Zykluszeit [ms]	$M_1-S_1-T_1$ [ms]	$M_2-T_2-S_2$ [ms]
160	25 – 55 – 80	25 – 25 – 110

Fehler bei der Nachrichtenübermittlung auf der Netzwerkschicht haben Paketverluste zur Folge. Diese werden nach Möglichkeit durch das  $MTS$ -Protokoll korrigiert. Jedoch müssen Wechselwirkungen durch den Einsatz der  $MST$ - $MTS$ -Konfiguration in der verwendeten Baumstruktur beachtet werden. Zur Evaluation der Reaktion auf Paketverluste werden in der Simulation bestimmte Pakete „verworfen“, also der Paketverlust gezielt initiiert. Es werden einmalige Paketverluste betrachtet. Wechselwirkungen oder Verzögerungen durch häufigere oder gleichzeitige Paketverluste sind nicht auszuschließen.

Der Verlust eines Slave-Beacons hat zum Beispiel zur Folge, dass die Pakete für die Registrierung der Slave-Knoten nicht gesendet werden. Die Registrierung wird dann in einem darauf folgenden Zyklus nachgeholt. Beim einmaligen Verlust kann die vollständige Kommunikation nach drei Zyklen, also im aktuellen Beispiel nach 480 ms, abgeschlossen werden. Das Master-Beacon markiert das Ende der Slave-Phase und dessen Verlust hat nur dann eine Auswirkung, wenn die Übertragungszeit der Slave-Pakete die Zeit der jeweiligen Slave-Phase übersteigt.

Beim Verlust eines Datenpaketes in der Master-Phase wird in der Token-Phase durch den Slave eine Meldung über den Nichterhalt der Nachricht zurückgegeben, sodass ein erneutes Senden in der nächsten Master-Phase initiiert wird. Somit wird die Kommunikation erst im dritten Zyklus abgeschlossen. Für die Registrierungsnachrichten werden Acknowledgements versendet. Geht eines dieser Pakete verloren, so nimmt der Slave-Knoten an, dass seine Statusnachricht den Master nicht erreicht hat. Somit wird das erneute Senden in der nächsten Slave-Phase ausgelöst. Auf das grundlegende Verhalten des Beispielmmodells in der Simulation hat dies keinen Einfluss, die Kommunikation wird innerhalb des zweiten Zyklus abgeschlossen.

Paketverluste in der Token-Phase können entweder das Token selbst oder die zugehörige Antwort betreffen. Geht das Token beim Versenden durch den Master verloren, so gibt es auch keine Rückmeldung durch den Slave-Knoten und es wird ein neues Token in der nächsten Token-Phase gesendet. Damit wird die Kommunikation um einen Zyklus verzögert. Der Verlust eines Antwortpaketes kann Wechselwirkungen durch die *MST-MTS*-Konfiguration auslösen. Durch Verlust eines Antwortpaketes vom  $STM_A$  kann die entsprechende weiterzugebende Antwort an den  $NCAP_I$  nicht vom  $NCAP_C$  bereitgestellt werden. Es erfolgt jedoch keine Rückmeldung. Dieser Ablauf zeigt, dass sich das Paket noch in Bearbeitung befindet, sodass im folgenden Zyklus ein neues Token durch den  $NCAP_I$  gesendet wird. Es wird aber kein erneutes Versenden der Nachricht durch den  $NCAP_I$  initiiert. Der ausgelöste Fehler im  $STM_A$  hat allerdings eine Aufforderung zum erneuten Versenden zur Folge. Dies ist erforderlich, da keine zusätzliche Bestätigung zu Token-Antworten versendet wird und die Datenpakete nicht mehrere Zyklen in den Slaves vorgehalten werden. Dieser Fehler macht zwei zusätzliche Kommunikationszyklen notwendig, sodass der Datenaustausch im aktuellen Beispiel erst nach 64 statt nach 32 ms abgeschlossen ist.

Eine verlorene Status-Nachricht in der Slave-Phase wird nicht durch ein Acknowledgement bestätigt, sodass genau wie im Falle eines Acknowledgement-Verlustes die Status-Nachricht erneut gesendet wird. Dieser Verlust hat jedoch

einen Einfluss auf die Kommunikation, indem beispielsweise die Registrierung nicht im ersten Zyklus abgeschlossen wird und somit die vollständige Kommunikation erst im dritten Zyklus durchgeführt ist.

### **Strategie zum Umgang mit Ausfall oder Fehlfunktion eines Knotens**

Fehler in der Hardware oder in der Software werden in der Slave-Phase an den entsprechenden Master mitgeteilt, sodass auf diese Fehler reagiert werden kann. Ausfälle oder Fehlfunktionen müssen für den jeweiligen Knotentypen einzeln beschrieben werden und werden im Folgenden exemplarisch behandelt.

Ein Fehler im  $STM_A$  hat im schlimmsten Fall, wenn zum Beispiel ein Aktuator durch einen mechanischen Defekt nicht mehr seiner Aufgabe nachkommen kann, den Ausfall eines kompletten Netzwerkbereichs zur Folge. Ist der Aktuator nicht mehr ansteuerbar aber sonst noch funktionsfähig, so kann die Aktuierung gegebenenfalls unter Anpassung der Parameter fortgesetzt werden. Für die Anwendung in der Strömungsregelung müssen die Stellgrößen dazu gegebenenfalls angepasst werden. Innerhalb des Netzwerkes ist dafür Sorge zu tragen, dass die Aktuierung bei einem Ausfall nicht die Mechanik des verwendeten Aktuator-Systems beschädigt. Die Behebung möglicher Hardware-Ausfälle in der analogen Ansteuerung können nur durch eine zusätzliche Redundanz abgedeckt werden. Die externe Strömungsregelung wird auf jeden Fall über einen etwaigen Aktuator-Ausfall informiert. Je nach Anwendung ist die Implementierung konkreter Strategien zur Kompensation eines solchen Fehlers vorgesehen.

Der Ausfall eines  $STM_S$  beziehungsweise aus anderen Gründen fehlerbehaftete Messungen sind durch entsprechende Signalverarbeitungsalgorithmen im  $NCAP_C$  nachweisbar und können somit in der weiteren Verarbeitung berücksichtigt werden. Wenn der Fehler lokal nicht ausgeglichen werden kann, wird diese Aufgabe an benachbarte Knoten oder, falls dies nicht möglich ist, an die jeweils höhere Ebene delegiert, bis gegebenenfalls die Strömungsregelung deaktiviert werden muss.

### 5.3.5 Bewertung der Evaluation des Netzwerkes und der Modellimplementierung

Der allgemeine Nachweis der korrekten Funktionsweise des Netzwerkes ist nicht durchführbar, sondern jeweils nur für eine konkret parametrisierte Netzwerkkonfiguration. Anhand einer vorgegebenen Knotenverteilung in einem Netzwerk mit sieben Teilnehmern wurde zunächst eine funktionierende Konfiguration ausgewählt. Diese Konfiguration konnte in einem weiteren Schritt durch die Optimierung der Zeitverteilung verbessert werden. Das zeigt die Möglichkeiten, die sich innerhalb Generierung, Konfiguration und Verwendung von Netzwerkmodellen bieten. Sofern die einzelnen Kommunikationsschichten aufeinander abgestimmt sind, ist eine funktionierende Netzwerksimulation zu erwarten. Dazu sind Parameter wie Knotenanzahl und Zeitverteilung auf der Transportschicht auszuwählen. Diese können anhand der Simulationsergebnisse und theoretischer Annahmen optimiert werden. Die Mechanismen zur Fehlerprävention und Fehlerbehandlung liefern zufriedenstellende Ergebnisse. Darauf aufbauend können nun verlässlich Aussagen für unterschiedliche Netzwerkkonfigurationen getroffen werden.

## 5.4 Simulationsbasierte Validierung der Aktuator- und Sensornetzwerke

Die Spezifikationen für das Netzwerk ergeben sich im Allgemeinen aus der Applikation und im speziellen aus den Anforderungen des entwickelten zentralen Regelungsalgorithmus. Allgemeine Anforderungen sind beispielsweise Energieeffizienz, Ausfallsicherheit oder auch Hardwareaufwand. Eine spezielle Anforderung ist zum Beispiel die notwendige Taktfrequenz des Reglers, also der maximal zu tolerierende Taktzyklus des Netzwerkes. Die Ende-zu-Ende-Verzögerung in dem Netzwerk bestimmt den minimal möglichen Taktzyklus für die über das Netzwerk umgesetzte zentrale Regelung und ist im Netzwerk durch Paketgrößen und Sendefrequenz sowie durch die Anzahl von Aktuatoren und Sensoren bestimmt. Zur Evaluierung, inwiefern die geforderten Eigenschaften qualitativ erfüllt werden, kann nun nach der erfolgreichen Validierung das Netzwerkmodell beziehungsweise die Simulationen konkreter Netzwerkkonfigurationen verwendet werden. Im Folgenden wird exemplarisch die Auswirkung der Konfiguration auf das Timing der Transportschicht sowie auf die Knotenverteilung und den Hardwarebedarf evaluiert.

## 5.4 Simulationsbasierte Validierung der Aktuator- und Sensornetzwerke

---

Es werden zwei verschiedene Beispiele betrachtet. Im ersten Beispiel werden homogene Strömungsbedingungen angenommen, sodass die Parameter für die  $STM_A$  als Broadcast vom  $NCAP_I$  an die  $NCAP_C$  verteilt werden. Das zweite Beispiel geht von inhomogenen Strömungsbedingungen aus, sodass die Stellgrößen-Parameter zu jedem  $NCAP_C$  einzeln verteilt und daher einzelne Kommunikationsphasen zusätzliche Zeit in Anspruch nehmen. Dies führt zu einer Verlängerung des Kommunikationszyklus. In beiden Fällen wird davon ausgegangen, dass über die Simulationszeit von jeweils 50 Kommunikationszyklen keine Fehlerbehandlung oder Synchronisation der Knoten notwendig ist. Außer den in Kapitel 3 beschriebenen Wartezeiten für Protokoll und Datenverarbeitung sind keine zusätzlichen externen Einflüsse oder Verzögerungen vorgesehen.

Es wird das schon in Abb. 3.10 vorgestellte Kommunikationsparadigma simuliert. Der Aufruf zur Datenerfassung zur vorgegebenen Zykluszeit erfolgt durch den  $NCAP_I$  über die  $NCAP_C$  an die  $STM_S$ . Die Verarbeitung der gemessenen Daten erfolgt im  $NCAP_C$  durch Mittelwertbildung aus den einzelnen empfangenen Werten der Sensoren. Zur Generierung einer Antwortnachricht an die externe Regelung für jeden Bereich werden die Sensordaten mit den verwendeten Aktuator-Parametern im  $NCAP_I$  zusammengeführt. Für die Simulation des Aktuator- und Sensornetzwerkes wird eine möglichst geringe Zykluszeit gesucht, sodass die externe Regelung mit einer hohen Taktfrequenz sowohl Messdaten erhält als auch die Stellgrößen verändern kann. Die Aktualisierung der Stellgrößen ist in jedem Zyklus möglich.

In beiden Konfigurationen (für die Annahme homogener und für die Annahme inhomogener Strömungsbedingungen) erfolgt die Anmeldung im ersten Zyklus. In Zyklus zwei wird die erste Aufforderung zur Bereitstellung von Sensordaten gesendet. In den nachfolgenden Zyklen werden dann sowohl Aktuierungs-Parameter gesetzt als auch Sensordaten abgefragt. Im homogenen Fall wird die Verteilung der Aktuierungs-Parameter per Broadcast und Multicast an die Slave-Knoten durchgeführt. Die Nachrichten an den Master-Knoten werden dabei direkt adressiert. Im inhomogenen Fall werden die Aktuierungs-Parameter an die einzelnen Bereiche, also an die  $NCAP_C$ , adressiert. Die Aufforderung zur Bereitstellung der Sensordaten erfolgt immer als Broadcast an alle  $NCAP_C$  beziehungsweise Multicast an die  $STM_S$ .

Für beide Fälle sind die Knotenverteilungen der verglichenen Netzwerke in Tabelle 5.11 aufgeführt. Dabei sind im ersten Netzwerk ( $N_1$ ) 144 Knoten zu 14 im  $NCAP_I$ - $NCAP_C$ -Netzwerk und jeweils 11 in den 13  $NCAP_C$ - $STM$ -Netzwerken zusammengefasst. Es sind 78  $STM_A$  und 52  $STM_S$  auf 13  $NCAP_C$  verteilt. Im zweiten Netzwerk ( $N_2$ ) sind 141 Knoten zu 11 im  $NCAP_I$ - $NCAP_C$ -

**Tabelle 5.11:** Knotenverteilung für zwei unterschiedliche evaluierte Netzwerke.

Name	1. Netzwerk ( $N_1$ )	2. Netzwerk ( $N_2$ )
NCAP <sub>I</sub>	1	1
NCAP <sub>C</sub>	13	10
STM <sub>A</sub> pro NCAP <sub>C</sub>	6	8
STM <sub>S</sub> pro NCAP <sub>C</sub>	4	5
Gesamt	144	141

Netzwerk und jeweils 14 in zehn NCAP<sub>C</sub>-STM-Netzwerken zusammengefasst. Darin sind 80 STM<sub>A</sub> und 50 STM<sub>S</sub> auf zehn NCAP<sub>C</sub> verteilt.

Die Simulation des Gesamtmodells bezieht alle Knoten in der Kommunikation mit ein. Die Analyse erfolgt hinsichtlich der Antwortzeit und der Rückmeldungen aus Sicht der externen Regelung. Es handelt sich um deterministische Simulationen ohne statistische Einflüsse, daher ist nur eine Simulation für jeden Parametersatz durchzuführen.

### 5.4.1 Simulation des Netzwerkes für homogene Strömungsbedingungen

Eine mögliche gesamte Zykluszeit von 210 ms wurde experimentell für  $N_1$  bestimmt. Zusammen mit weiteren Simulationen erlaubt dies die Bestimmung einer Zykluszeit für  $N_2$  von 180 ms. Die genaue Zeitverteilung der einzelnen Kommunikationsphasen ist Tabelle 5.12 zu entnehmen.

Die Simulationen beider Netzwerkkonfigurationen verlaufen ohne Fehler, das heißt die Randbedingungen des Protokollstapels werden nicht verletzt. Die Antworten der Nachrichten sind am Ende der einzelnen Zyklen jeweils vollständig abgearbeitet und für die externe Regelung verfügbar. In Abb. 5.16 lassen sich beide Netzwerkkonfigurationen  $N_1$  (siehe Abb. 5.16a) und  $N_2$  (siehe Abb. 5.16b) anhand der Kommunikation des NCAP<sub>I</sub> mit der externen Regelung vergleichen. Die gesendete und empfangene ID bezeichnet jeweils

## 5.4 Simulationsbasierte Validierung der Aktuator- und Sensornetzwerke

**Tabelle 5.12:** Zeitverteilung der einzelnen Kommunikationsphasen in der Transportschicht für homogene Strömungsbedingungen in zwei unterschiedliche Netzwerkkonfigurationen.

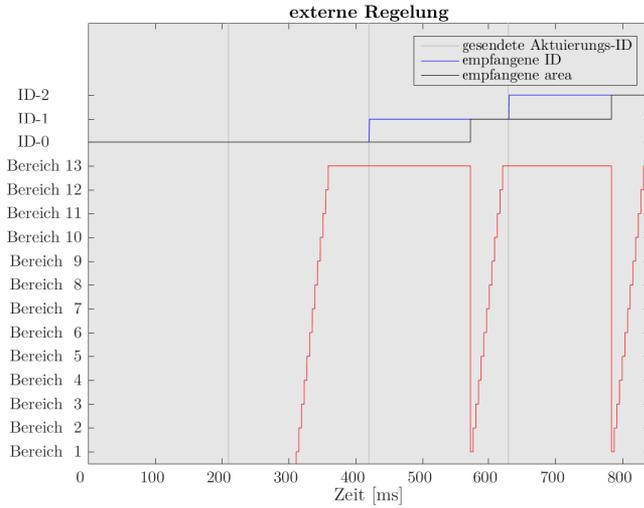
Netzwerk	Zykluszeit [ms]	$M_1-S_1-T_1$ [ms]	$M_2-T_2-S_2$ [ms]
$N_1$	210	50 – 50 – 110	50 – 30 – 130
$N_2$	180	50 – 50 – 80	50 – 30 – 100

den Ausgang und den Eingang von Nachrichten. Der Empfang des **area**-Parameters beschreibt, aus welchen Bereichen die Antwortnachrichten eintreffen. Neue Aktuierungs-Parameter werden nach dem Eintreffen der Antworten gestellt. Die Überdeckung von  $S_1$  und  $S_2$  wird zur Datenverarbeitung verwendet, sodass am Ende der Phase  $T_2$  noch nicht alle Pakete in  $\text{NCAP}_C$  verarbeitet sind (siehe Tabelle 5.12).

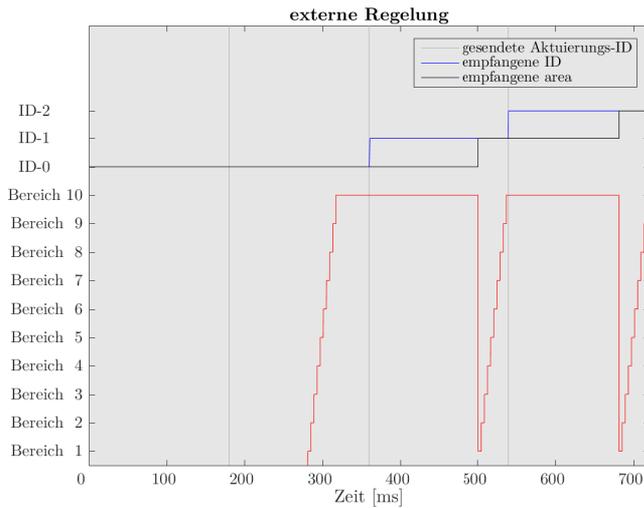
Die Antwortzeit in  $N_1$  zwischen dem Ausgang der Stellgrößen und der Beantwortung dieser mit den Datenpaketen, welche die umgesetzten Parameter und erfasste Messwerte enthalten, beträgt 152 ms. Bis alle Nachrichten eingetroffen sind, vergehen weitere 48 ms. Es vergehen also 200 ms bis zur vollständigen Beantwortung der Nachrichten durch das Netzwerk. Im Netzwerk  $N_2$  dauert es 141 ms bis zur ersten Antwort und weitere 36 ms bis die Antworten für alle Bereiche eingetroffen sind. Insgesamt dauert die Beantwortung 177 ms.

Die Netzwerkauslastung beträgt in  $N_1$  im  $\text{NCAP}_I\text{-NCAP}_C$ -Netzwerk 5,33% und 2,43% innerhalb des  $\text{NCAP}_C\text{-STM}$ -Netzwerkes. In  $N_2$  liegt die Netzwerkbelastung bei 5,46% im  $\text{NCAP}_I\text{-NCAP}_C$ -Netzwerk und 4,06% im  $\text{NCAP}_C\text{-STM}$ -Netzwerk. Gegenüber  $N_1$  ist in  $N_2$  ein Teil des Datenverkehrs in die Subnetzwerke verschoben. Zur physikalischen Vernetzung werden für  $N_1$  157 und für  $N_2$  151 Kabelverbindungen benötigt.

Obwohl es nur geringe Unterschiede in der Konfiguration und der Knotenanzahl beziehungsweise -verteilung gibt, weist das zweite Netzwerk eine geringere Zykluszeit beziehungsweise Antwortzeit auf. Der Vergleich von  $N_1$  und  $N_2$  zeigt, dass weniger Verbindungen für das zweite Netzwerk notwendig sind. Das Netzwerk  $N_2$  stellt also eine hardware- und softwareseitige Verbesserung durch minimale Variation der Modellparameter dar und bietet eine bessere Auslastung der Netzwerkinfrastruktur.



(a) Darstellung der Kommunikation zwischen externer Regelung und NCAP<sub>I</sub> in  $N_1$ .



(b) Darstellung der Kommunikation zwischen externer Regelung und NCAP<sub>I</sub> in  $N_2$ .

**Abb. 5.16:** Vergleich der beiden evaluierten Netzwerke für inhomogene Strömungsbedingungen. Der Datenaustausch stimmt überein und wird in beiden Simulationen korrekt durchgeführt. Die Zykluszeit in  $N_1$  beträgt jedoch 210 ms und in  $N_2$  180 ms.

Wird die Zeitverteilung von  $N_2$  in  $N_1$  verwendet, so kann die Kommunikation nicht vollständig innerhalb der Kommunikationszyklen abgeschlossen werden. Die Datenpuffer für eingehende Nachrichten können nicht geleert und somit die entsprechenden Daten der externen Regelung nicht zur Verfügung gestellt werden.

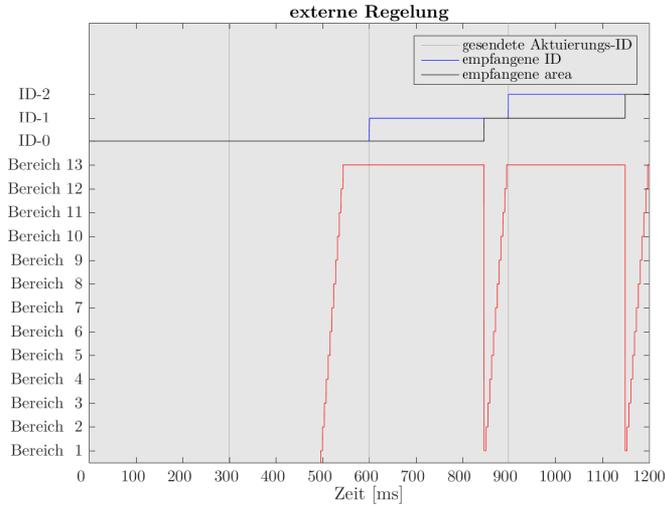
### 5.4.2 Simulation des Netzwerkes für inhomogene Strömungsbedingungen

Bei der Annahme inhomogener Strömungsbedingungen für die externe Regelung muss für die Versendung der Pakete in der Master-Phase mehr Zeit zur Verfügung gestellt werden. Die zeitliche Überdeckung der Slave-Phasen sollte hierfür nicht genutzt werden, damit jede Kommunikationsphase vor Beginn der nächsten Phase abgeschlossen ist. Dies vermeidet Folgefehler im Falle vieler verschiedener Datenpakete.

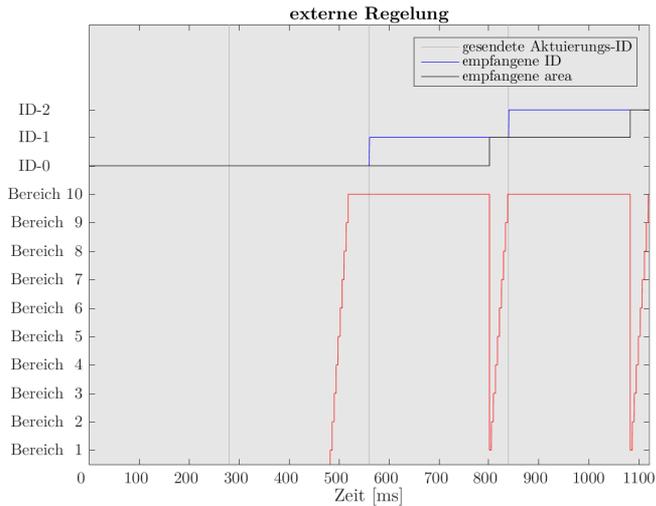
Die Bestimmung der Zykluszeiten entsprechend der in Kapitel 5.3.2 vorgestellten Berechnung liefert für  $N_1$  zunächst eine minimale Zykluszeit von 236 ms ohne Berücksichtigung der Registrierung. Die Zykluszeit ist verteilt auf 63 ms für die erste, 74 ms für die zweite und 95 ms für die dritte Phase. Die benötigte Zeit des Registrierungsprozesses beträgt 39 ms für die Anmeldung der STM an die NCAP<sub>C</sub> und 49 ms für die Anmeldung der NCAP<sub>C</sub> an den NCAP<sub>1</sub>. In  $N_2$  sind es 222 ms: für die erste 53 ms, für die zweite 95 ms und für die dritte Phase 74 ms. Die Anmeldung an die NCAP<sub>C</sub> benötigt 49 ms und die Anmeldung an die NCAP<sub>1</sub> 39 ms.

Mit diesen Werten werden, wie in Kapitel 5.3.2 dargestellt, die Phasen  $S_1$  und  $S_2$  angepasst. Die daraus resultierenden Zykluszeiten ergeben keine funktionierende Netzwerkkonfiguration, da die Nachrichtenübertragung bei der Berechnung als parallel angenommen wird. In der Simulation ist bei detaillierter Analyse der Simulationsergebnisse zwischen den einzelnen Antworten ein geringer zeitlicher Versatz von 1,1 ms erkennbar. Es wird eine Anpassung der Zeitverteilung vorgenommen, sodass für  $N_1$  300 ms und für  $N_2$  280 ms resultieren. Die genaue Verteilung jeweils für eine funktionierende Konfiguration ist in Tabelle 5.13 beschrieben.

Die Applikation unterscheidet sich vom homogenen Fall insofern, dass mehrere Eingangsnachrichten abgearbeitet werden und an jeden Bereich einzeln weitergegeben werden müssen. Mit der Konfiguration aus Tabelle 5.13 sind beide Netzwerke stabil und verletzen keine Randbedingungen der Kommunikation. Auch die Abarbeitung aller Pakete ist innerhalb der Zyklen gewährleistet.



(a) Darstellung der Kommunikation zwischen externer Regelung und  $\text{NCAP}_I$  in  $N_1$ .



(b) Darstellung der Kommunikation zwischen externer Regelung und  $\text{NCAP}_I$  in  $N_2$ .

**Abb. 5.17:** Vergleich der beiden evaluierten Netzwerke für inhomogene Strömungsbedingungen. Der Datenaustausch stimmt überein und wird in beiden Simulationen korrekt durchgeführt. Die Zykluszeit in  $N_1$  beträgt jedoch 300 ms und in  $N_2$  280 ms.

**Tabelle 5.13:** Zeitverteilung der einzelnen Kommunikationsphasen in der Transportschicht für inhomogene Strömungsbedingungen in zwei unterschiedliche Netzwerkkonfigurationen.

Netzwerk	Zykluszeit [ms]	$M_1-S_1-T_1$ [ms]	$M_2-T_2-S_2$ [ms]
$N_1$	300	67 – 128 – 105	67 – 89 – 144
$N_2$	280	57 – 144 – 89	57 – 105 – 128

Die Antworten der gesendeten Stellgrößen sind innerhalb des Zyklus vollständig abgearbeitet und für die externe Regelung verfügbar.

In Abb. 5.17 lässt sich das erste Netzwerke  $N_1$  (siehe Abb. 5.17a) mit dem zweiten Netzwerk  $N_2$  (siehe Abb. 5.17b) anhand der Kommunikation mit der externen Regelung vergleichen. Der Änderung der ID wird innerhalb des Zyklus gefolgt und durch den Empfang der jeweils 13 beziehungsweise zehn verschiedenen **area**-Parameter abgeschlossen. Neue Aktuierungs-Parameter werden nach dem Eintreffen der Antworten gestellt. Die Antwortzeit in  $N_1$  zwischen dem Ausgang der Stellgrößen und der ersten Antwort beträgt 249 ms. Bis zum Eintreffen aller Antworten vergehen weitere 48 ms. Im Netzwerk  $N_2$  wartet die externe Regelung 242 ms bis zur ersten Antwort und weitere 36 ms bis die Antworten für alle Bereiche eingetroffen sind. Insgesamt dauert die Beantwortung 278 ms.

Die Netzwerkauslastung in  $N_1$  liegt bei 3,64% im NCAP<sub>I</sub>-NCAP<sub>C</sub>-Netzwerk und 1,36% im NCAP<sub>C</sub>-STM-Netzwerk. Im zweiten Netzwerk  $N_2$  beträgt sie 2,83% für die Datenübertragung zwischen NCAP<sub>I</sub> und NCAP<sub>C</sub> und 2,13% für die Verbindung zwischen NCAP<sub>C</sub> und STM.

Wie schon im homogenen Fall bietet das Netzwerk  $N_2$  auch im inhomogenen Fall eine kürzere Zykluszeit und somit eine kürzere Antwortzeit. Außerdem müssen bei der physikalischen Realisierung weniger Kabel verwendet werden und auch die Auslastung des Netzwerkes ist insgesamt höher, sodass die Ressourcen effizienter genutzt werden. Jedoch lässt  $N_2$  in beiden Fällen weniger Spielraum für Kommunikationsfehler, da die Kommunikation wenige Millisekunden vor dem Ende eines vollständigen Zyklus abgeschlossen ist und neue Stellgrößen direkt danach an das Netzwerk übertragen werden.

### 5.4.3 Bewertung der Simulationen

Der Vergleich verschiedener Netzwerkkonfigurationen zeigt die Möglichkeiten der flexiblen Generierung und Untersuchung der Netzwerkmodelle. Dabei lassen sich Unterschiede sowohl für die Konfiguration, als auch für die externe Regelung aufzeigen. Je nach Anwendung ist ein geringer Unterschied in der Anzahl von Aktuatoren und Sensoren oder Kabelverbindungen beispielsweise aus Kostengründen entscheidend. Für zeitkritische Anwendungen mit einer möglichst schnellen Sensorabtastung kann ein Zeitunterschied von fünf Millisekunden bereits bedeutend sein. Eine erweiterte Bewertung der Netzwerkkonfigurationen hinsichtlich des Energieverbrauchs ist möglich, sobald der Energieverbrauch für die Aktuierung und die Anforderungen an die Rechenleistung der zentralen Regelung und der Netzwerkknoten abgeschätzt werden können. Dies ist jedoch erst nach abgeschlossener Entwicklung der robusten Methoden zur Widerstandsreduktion durch transversale Oberflächenwellen möglich.

## 5.5 Steuerung und Regelung des Aktuator-Systems

In diesem Kapitel wird die Funktionsweise der Erzeugung der wandernden Oberflächenwellen auf einer Aluminiumplatte validiert. Anhand von Referenzmessungen werden Offset-, Amplituden-, Frequenz- und Phasenänderungen geprüft. Zudem wird ein Ansatz zur modellbasierten Entwicklung einer robusten Wellenregelung vorgestellt.

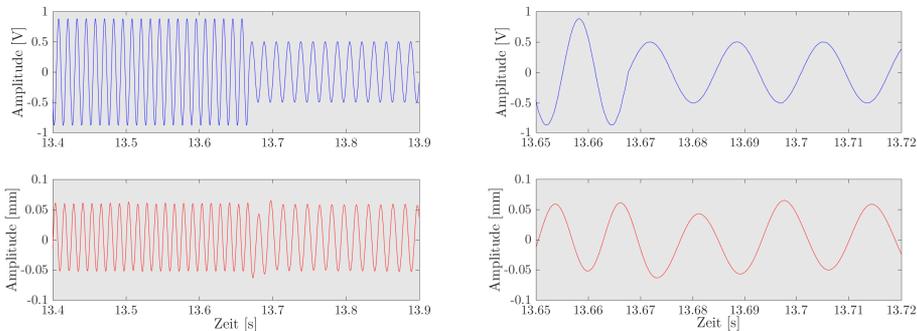
### 5.5.1 Echtzeitige Generierung und Änderung der transversalen Oberflächenwellen

Die Generierung und Änderung der transversalen Oberflächenwellen wird über die Messung des Ausgangssignals des DAQ-Systems und einer Oberflächenmessung mit einem Punkt-Lasersystem von KEYENCE [48] validiert. Die Software-Tests und Referenzmessungen wurden sowohl mit dem neuen Aktuator-System, als auch mit einem Prototypenaufbau mit drei Balken und einem Balkenabstand von zehn Millimetern durchgeführt. Um die Einflüsse

der nichtlinearen Kopplung der Aktuatoren über die Aluminiumoberfläche auszuschließen wird nur ein einzelner Aktuator-Balken bewegt.

Bei der Aktuierung durch nur einen Balken mit einer Frequenz von 40 Hz und einem Millimeter Amplitude wird bei einer Steuerspannung von 0,5 V eine effektive Spannung von 10,3 V und ein Strom von 560 mA an der Aktuator-Spule gemessen.

$$x_0 = \frac{\frac{F_1}{m}}{\sqrt{(\omega_0^2 - \omega_1^2)^2 + \left(\frac{\omega_1 b}{m}\right)^2}} \quad (5.7)$$

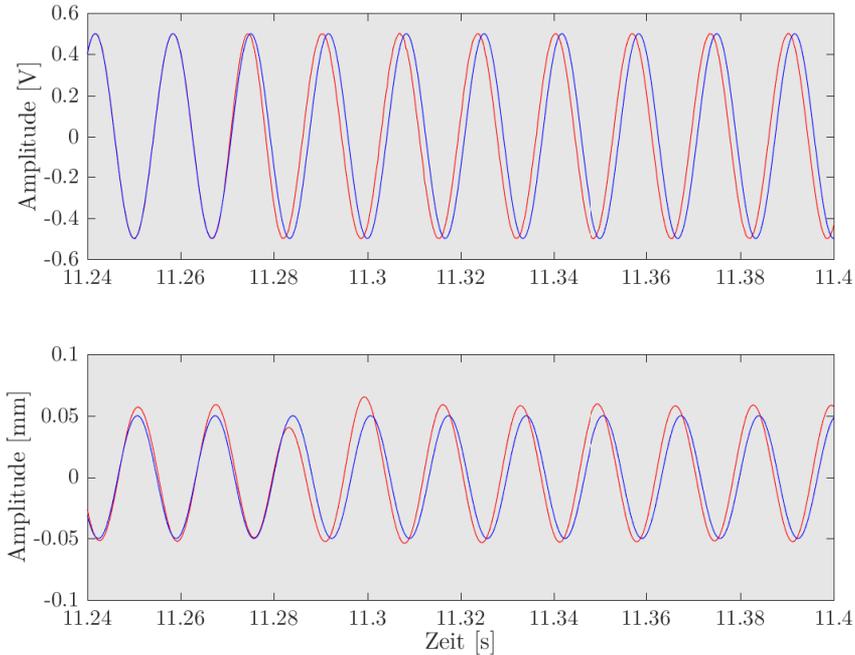


- (a) Der Parameterwechsel in Frequenz und Amplitude beschränkt sich auf etwa eine Periode in der Balkenbewegung.      (b) Nahaufnahme des Parameterwechsels in Frequenz und Amplitude mit glattem Signalübergang.

**Abb. 5.18:** Analogausgang (blau, oben) des DAQ-Systems und Amplitudenverlauf (rot, unten) eines einzelnen angeregten Aktuator-Balkens. Der Parameter-Übergang erfolgt von 80 Hz mit 0,88 V zu 60 Hz mit 0,5 V, sodass die Amplitude der Oberflächenbewegung nahezu konstant bleibt.

Die Änderung des Offsets beeinflusst die Höhe der Aktuatoren sowohl in der Ruhelage als auch während der Generierung der Oberflächenwellen. Sie erfolgt jedoch auf einer langsameren Zeitskala, da sie nicht direkt an das Ausgangssignal gekoppelt ist. In der Ruhelage ist eine Offsetanpassung der Aluminiumplatte von  $\pm 200 \mu\text{m}$  möglich.

Abb. 5.18a zeigt den Wellenverlauf bei einem gleichzeitigen Wechsel der Wellenfrequenz  $F$  und der Wellenamplitude  $A$ . Dabei sind das Ansteuerungssignal im oberen Graph und das gemessene Signal im unteren Graph aufgetragen. Die automatisiert berechnete glatte Transiente ist in Abb. 5.18b hervorgehoben.



**Abb. 5.19:** Die Messung der Phasenänderung um  $30^\circ$  der analogen Anregung (rot, oben) und Reaktion eines einzelnen angesteuerten Aktuator-Balkens (rot, unten). Zum Vergleich ist eine Sinuswelle mit konstanten Parametern als Referenzsignal hinzugefügt (jeweils blau).

Der Übergang zwischen den Wellenparametern erfolgt innerhalb einer Periode. Die Amplitude des Anregungssignals wurde in diesem Beispiel angepasst, sodass sie die für getriebene gedämpfte Oszillatoren typische Abhängigkeit der resultierenden Amplitude von der Anregungsfrequenz möglichst genau kompensiert. Die Gleichung 5.7 beschreibt diese Wechselwirkung,  $F_1$  bezeichnet die Eingangsamplitude und  $\omega_1$  die Eingangsfrequenz. Die Eigenfrequenz des

Systems wird durch  $\omega_0$  beschrieben. Die Dämpfung und das Gewicht des getriebenen Oszillators fließen durch  $b$  und  $m$  ein.

Abb. 5.19 zeigt den Wellenverlauf bei einer Phasenänderung um  $30^\circ$  im laufenden Betrieb. Auch hier ist der Übergang von der alten zur neuen Phasenlage innerhalb einer Periode abgeschlossen. Zur besseren Vergleichbarkeit ist ein Referenzsignal ohne Phasenverschiebung eingezeichnet.

Die verwendete Software ermöglicht einen minimalen Updatezyklus des Ansteuerungs-Signals von  $20 \mu\text{s}$  auf dem Echtzeitsystem, wobei noch Ressourcen für die vorgesehene Wellenregelung vorgehalten sind.

### 5.5.2 Modellbasierte Entwicklung der Wellenregelung (innerer Regelkreis)

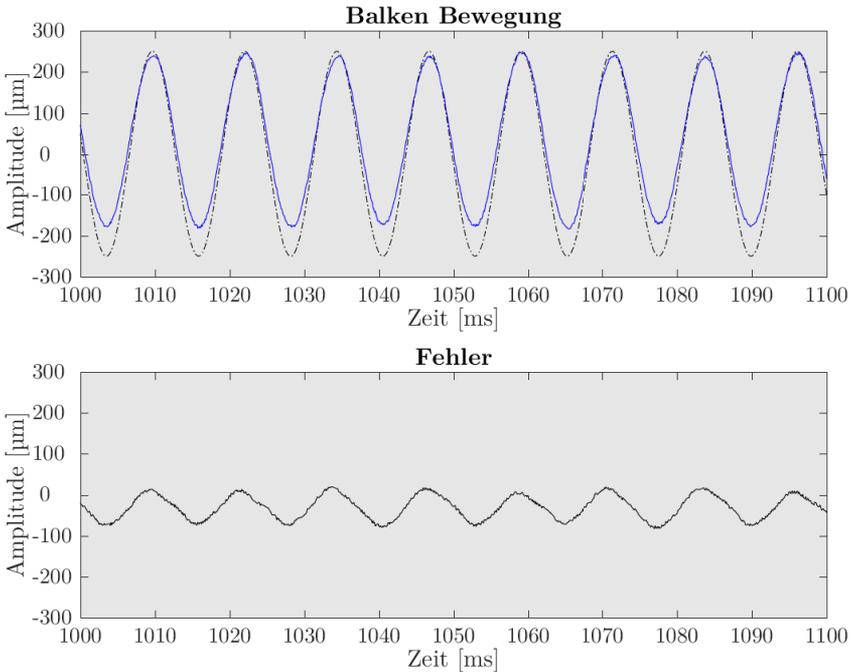
Im Folgenden wird die experimentelle Anwendung klassischer Regler an zunächst zur einem Aktuator-Balken evaluiert. Hierfür wird der im vorigen Kapitel beschriebene reduzierte Prototyp des neuen elektromagnetischen Aktuator-Systems verwendet, der über MATLAB/Simulink Real-Time angesprochen wurde. Als kontinuierliches Eingangssignal wurde das Signal des KEYENCE Laser Systems [48] für die Bestimmung der Amplitude eingelesen.

Das bisher in Kapitel 4.4 verwendete mathematische Modell des Aktuator-Systems approximiert das Verhalten der Aktuator-Balken unter Berücksichtigung der Materialkonstanten. Nicht alle Parameter des Systems, die für die Entwicklung des Reglers notwendig sind, sind einer direkten Messung zugänglich. Daher wird eine Identifikation des Systemverhaltens mit Hilfe der System Identification Toolbox von MATLAB durchgeführt, siehe auch [1].

Zur Systemidentifikation wird zunächst die Dynamik eines einzelnen Balkens abgebildet. Unter der Annahme, dass sich die Dynamik der einzelnen Aktuator-Balken durch ein  $PT_2$ -Modell mit  $D < 1$  darstellen lässt (Notation aus Abel [2]), wird die durch die angelegte Spannung erzeugte Bewegung eines einzelnen Balkens mit der Übertragungsfunktion  $F_S$  in Gleichung 5.8 beschrieben [30].

$$F_S(s) = \frac{0,187 \frac{\text{mm}}{\text{V}}}{3,37 \cdot 10^{-5} \cdot s^2 + 0.012 \cdot s + 1} \quad (5.8)$$

Die Toolkette zur experimentellen Validierung verschiedener Strategien der Wellenregelung am Prototyp des Aktuator-System wurde implementiert und getestet. Die Regelung kann am Modell evaluiert und über die Simulink Real-Time-Toolbox auf das Aktuator-System übertragen und validiert werden. Dabei sind insbesondere die speziellen Charakteristiken des Testsystems zu beachten, wie zum Beispiel die noch fehlende Ansteuermöglichkeit der Offsetanpassung. Das System besitzt eine Totzeit und das Stellsignal ist proportional zur Beschleunigung des Aktuator-Balkens. Somit ergibt sich im eingeschwungenen Zustand eine Phasenverschiebung zwischen Stellsignal und Messgröße. Diese wird im Folgenden nicht weiter betrachtet, kann aber für die Implementierung der vorgesehenen Regelstrategie ausgenutzt werden.



**Abb. 5.20:** Die Anwendung des PD-Reglers auf die durch einen Aktuator-Balken bewegte 0,3 mm starke Aluminiumoberfläche (blau) sowie die gewünschte Trajektorie (gestrichelt) sind im oberen Graphen dargestellt. Der Regler stabilisiert das System und generiert eine bleibende Regelabweichung.

## 5.6 Ankopplung an die Strömungsregelung (Model-in-the-Loop-Simulation)

---

Ein nach dem Hurwitz-Kriterium [2] ausgelegter PD-Regler mit angepasstem Reaktionsverhalten zeigt für die Aktuierung der Aluminiumoberfläche das in Abb. 5.20 veranschaulichte Verhalten. Der Fehler ist im unteren Graphen abgebildet.

Für die Aktuierung des Balkens mit einer Amplitude von  $250\ \mu\text{m}$  und einer Frequenz von 81 Hz wird bedingt durch die fehlende Offsetanpassung die vorgegebene Genauigkeit von fünf Prozent nur bei positiver Auslenkung erreicht. Im negativen Bereich ist die Abweichung größer als zehn Prozent. Der Aktuator wird insgesamt statt  $500\ \mu\text{m}$  nur  $450\ \mu\text{m}$  ausgelenkt.

Die Anwendung des PD-Reglers resultiert in einer bleibenden Regelabweichung. Die Einbeziehung der Rückstellkräfte zeigt bereits am Modell bei der Betrachtung als nichtlineare zeitlich variante Störung für die Bewegung eines einzelnen Balkens, dass konventionelle Regler mit realistischen Verstärkungsfaktoren das System nicht regeln können.

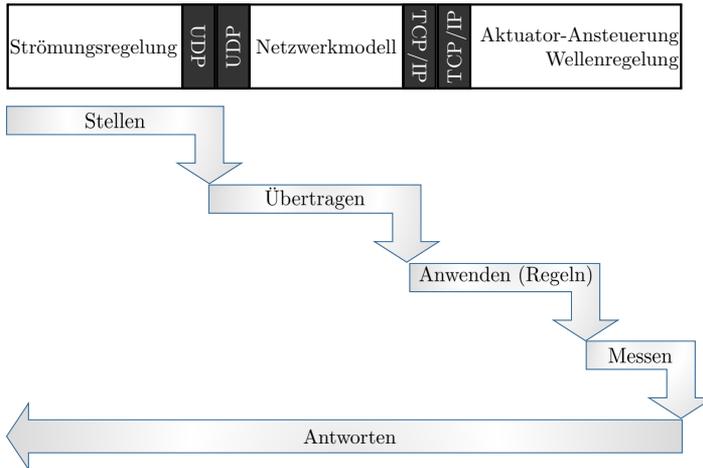
Die Systemidentifikation bestätigt die in Kapitel 4.3 vorgestellte DGL für das Aktuator-System und spezifiziert die notwendigen Modellparameter. Das Experiment am Aktuator-System zeigt, dass ein lernender Regler, wie er in Kapitel 4.4 oder auch in der Literatur [89] vorgestellt wird, einen vielversprechenderen Ansatz zur Regelung des Systems bietet. Für den Einsatz in der Strömungsregelung steht die Untersuchung verschiedener Ansätze, wie die Iterativ Lernende Regelung oder zum Beispiel die Repetitive Regelung (engl.: Repetitive Control, RC) nicht nur hinsichtlich Funktionalität und Regelgüte, sondern auch Energieeffizienz noch aus.

## 5.6 Ankopplung an die Strömungsregelung (Model-in-the-Loop-Simulation)

In der Model-in-the-Loop-Simulation wird das Netzwerkmodell mit der externen Strömungsregelung und dem bisher verwendeten Prototypenaufbau des Aktuator-Systems über die definierten Schnittstellen (siehe Kapitel 3.3.4) aus einer XML-basierten UDP und einer ASCII-basierten TCP/IP Netzwerkverbindung gekoppelt.

Der Ablauf ist in Abb. 5.21 dargestellt. Die externe Strömungsregelung sendet die gewünschten Stellgrößen an die Netzwerkschnittstelle. Die Werte werden innerhalb des Netzwerkes weitergeleitet und an die Aktuator-Ansteuerung

übergeben. Die Parameter werden verwendet um das Ausgabesignal anzupassen und ein Sinusförmiges Analogsignal zur Ansteuerung der Aktuatorik zu generieren. Tatsächliche Werte für die angewendeten Parameter und die Ergebnisse zur Messung der Widerstandsreduktion werden dann in einer Antwort zurückübertragen.



**Abb. 5.21:** Die Stellgrößen der externen Regelung werden über die Schnittstelle durch die Model-in-the-Loop-Simulation des Netzwerkes an die Aktuator-Ansteuerung übertragen. Sensordaten werden dort aufgenommen und zurück durch die Schnittstellen an die Strömungsregelung geleitet.

In einem Experiment mit einem reduzierten Netzwerkmodell wurden diese Schnittstellen validiert. Die Verwendung eines reduzierten Netzwerkes ermöglicht die Simulation in Echtzeit. Für große Netzwerksimulationen wird neben der Verzögerung durch die simulierte Datenübertragung eine zusätzliche Verzögerung induziert, da die Netzwerksimulation dann langsamer arbeitet als der externe Regler und die Aktuator-Ansteuerung. Für die Validierung der Schnittstellen ist dies jedoch nicht relevant. Die implementierten Schnittstellen werden sowohl für die Parameterstudien als auch für die Verwendung der externen Strömungsregelung im Windkanal eingesetzt. Somit wird der äußere Regelkreis der Kaskadenregelung durch das Netzwerk geschlossen.

## 5.7 Zusammenfassung

Die vorgestellte Modellimplementierung in MATLAB/Simulink und der TrueTime-Toolbox wird zur Generierung konkreter Modellkonfigurationen für die anschließende simulationsbasierte Validierung des Netzwerkes verwendet. Bei der Entwicklung mikrocontrollerbasierter Netzwerke für die Anwendung in der Strömungsregelung muss bei bekannten Anforderungen der externen Regelung das Netzwerkmodell entsprechend konfiguriert und mögliche Rechenzeiten müssen sowohl auf der Seite der externen Regelung als auch auf der Seite der Aktuierung und Sensorik berücksichtigt werden.

Die Funktionsweise des Modells ist anhand des beschriebenen Konzeptes umgesetzt. Ein physikalisches Beispielnetzwerk mit wenigen Knoten ermöglicht die Bestimmung und Validierung realitätsnaher Parameter. Da die Simulation und die Testumgebung für eine vergleichbare Konfiguration übereinstimmen, schafft dies Sicherheit, um mittels Simulationen hochskalierter Netzwerkmodelle verlässliche Vorhersagen für eine spätere physikalische Realisierung zu treffen. Bei der Generierung und Simulation von Netzwerkmodellen müssen Beschränkungen zur Limitierung der Anzahl und Verteilung der Knoten im Netzwerk beachtet werden. Ein wichtiger Aspekt ist zudem die Abstimmung der einzelnen Komponenten im Aktuator- und Sensornetzwerkmodell. Für kleine Netzwerke lassen sich sowohl konservative Annahmen für die Verteilung der *MST-MTS*-Zykluszeiten treffen als auch analytische Optimierungen vornehmen.

In einem Beispiel wurden ausgedehnte Netzwerke mit einer hohen Anzahl Knoten aus den einzelnen Gruppen  $\text{NCAP}_I$ ,  $\text{NCAP}_C$ ,  $\text{STM}_A$  und  $\text{STM}_S$  zusammen mit den validierten Netzwerkparametern anhand vorgegebener Randbedingungen verglichen. Dabei wurden sowohl homogene als auch inhomogene Randbedingungen betrachtet. Für die meisten Anwendungen erscheint es sinnvoll, Messung und Aktuierung zu trennen und die Datenübertragung in verschiedenen Zyklen auszuführen. Das Netzwerkmodell verhält sich dann robust gegenüber Einflüssen, wenn nicht nur die minimale *MST-MTS*-Zykluszeit für die entsprechenden Pakete in der Anwendung geplant wird. Die Synchronisation im Netzwerkmodell stellt nur die Kommunikation sicher, nimmt jedoch zunächst keine Rücksicht auf die spätere Anwendung des Netzwerkes. Die Anpassung der Switch-Puffergröße oder die Verwendung einer anderen Kommunikationsstrategie auf der Netzwerkschicht ermöglicht die Untersuchung größerer unterschiedlicher Netzwerke mit verschiedenen Paradigmen auf den beiden Ebenen der Baumstruktur. Diese sind durch das implementierte *CreNeMo*-Werkzeug schnell und einfach realisierbar.

Die vorgestellte echtzeitige Ansteuerung der Aktuatoren mit automatisiert berechneten glatten Transienten zwischen unterschiedlich parametrisierten Oberflächenwellen ermöglicht eine effiziente Durchführung der notwendigen Parameterstudien und der Strömungsregelung im Windkanal. Die Aktuator-Ansteuerung verwendet für die Erzeugung unterschiedlich parametrisierter Oberflächenwellen einen Datensatz einmalig berechneter Wellenformen. Das flexible Auslesen dieser Wertetabellen garantiert den Wechsel zwischen unterschiedlich parametrisierten Wellen innerhalb einer Periode. Dabei wird eine möglichst glatte Transiente berechnet, um den Eintrag hochfrequenter Störungen in das Strömungsfeld zu vermeiden.

Die robuste Regelung der Oberflächenwellen durch die Bewegung der Aktuator-Balken erfordert eine genauere Modellierung des Systemverhaltens und die Auswahl eines geeigneten Regelungskonzeptes hinsichtlich Regelgüte und Energieeffizienz. Im Gegensatz zu dem im Modell getesteten Ansatz eines Iterativ Lernenden Reglers ist die Verwendung eines klassischen PID-Reglers nicht erfolgversprechend. Der kaskadierte Regelkreis bestehend aus externer Regelung, Netzwerksimulation und aktuatornaher Regelung wird mithilfe der definierten Schnittstellen geschlossen.

## 6 Diskussion und Ausblick

In dieser Arbeit wurde ein echtzeitfähiges Aktuator- und Sensornetzwerk für die räumlich ausgedehnte Umsetzung einer auf transversalen Oberflächenwellen basierenden Strömungsregelung entwickelt.

Mithilfe des Netzwerkes und des angeschlossenen elektromagnetischen Aktuator-Systems werden in Form einer Model-in-the-Loop-Simulation die Parameterstudien im Windkanal zur Entwicklung der robusten Methoden für die Widerstandsreduktion durchgeführt.

Damit sind hinsichtlich der Umsetzbarkeit einer aktiven Widerstandsreduktion durch transversale Oberflächenwellen zwei wichtige Nachweise geführt worden. Zum einen können die räumlich ausgedehnt verteilten Aktuatoren und Sensoren mit einem robusten und echtzeitfähigen Netzwerk an die zentrale Regelung angebunden werden. Zum anderen erlaubt der mit dem Netzwerk realisierte kaskadierte Regelkreis die geregelte Erzeugung der transversalen Oberflächenwellen mit einem elektromagnetischen Aktuator-System. Als Oberflächenmaterial zur Wellenerzeugung wird ganz bewusst Aluminium verwendet, wie es beispielsweise heute in der Luftfahrt eingesetzt wird.

Die Variation der Oberflächenwellen entsprechend der Vorgaben der zentralen Regelung wird durch die aktuatornahe Berechnung möglichst glatter Transienten in Echtzeit innerhalb nur einer Wellenperiode umgesetzt. Mit den so betriebenen Windkanalversuchen konnte innerhalb der Forschergruppe FOR 1779 eine Reduktion des Reibungswiderstandes durch transversale Oberflächenwellen um 3,4% bei Verwendung einer glatten Aluminiumoberfläche und um sogar 9,4% bei Verwendung einer Riblet-strukturierten Oberfläche nachgewiesen werden [72][52].

Die Modellimplementierung des entwickelten Netzwerkes ist sehr flexibel ausgelegt. Das Gesamtmodell ermöglicht die Simulation großer Netzwerke unterschiedlicher Topologien und komplexer Kommunikationsparadigmen. Ausgehend von der definierten Aufgabenverteilung können verschiedene Modellkonfigurationen schnell und einfach erzeugt werden. Für unterschiedliche Ansätze

für die Strömungsregelung können so über entsprechende Simulationen optimierte Netzwerkkonfigurationen bestimmt werden. Neben der Reaktionszeit, Robustheit und dem Hardwareaufwand kann auch der Energiebedarf für die lokalen Berechnungen, die physikalische Aktuierung und die Netzwerkkommunikation für die Optimierung berücksichtigt werden.

Nach Abschluss der Entwicklung der robusten Methoden zur Strömungsbeeinflussung erlauben die Modellsimulationen die Spezifikationen der Netzwerkknoten und übrigen Netzwerkkomponenten für die physikalische Realisierung eines ersten Prototyps des echtzeitfähigen Aktuator- und Sensornetzwerkes. Als Basis für die Netzwerkknoten steht die im ZEA-2 entwickelte iNODE-Plattform zur Verfügung, die entsprechend der Spezifikationen flexibel erweitert werden kann.

Die vorliegende Modellimplementierung erlaubt auch die Weiterentwicklung des Netzwerkes hinsichtlich der geänderten Anforderungen durch die Betrachtung instationärer Anströmungsbedingungen in Kombination mit einem positiven Druckgradienten, wie dies in der zweiten Förderperiode der Forschergruppe vorgesehen ist. Hierfür müssen insbesondere die Reaktionszeit innerhalb des kaskadierten Regelkreises verkürzt werden. Diese setzt sich zusammen aus der Zeit für die Kommunikationsübertragung und der Zeitkonstante der Iterativ Lernenden Regelung im inneren Regelkreis.

Die Kommunikationszeiten können durch die Integration weiterentwickelter Protokollstandards verkürzt werden. So erlaubt der IEEE 802 Standard für Ethernet-basierte Netzwerke bis zu 100 GBit/s. Unter Berücksichtigung der einzusetzenden verteilten Signal- und Datenverarbeitungsalgorithmen kann das middlewarebasierte *MTS*-Übertragungsprotokoll weiter modifiziert werden. So kann die softwarebasierte echtzeitige Kommunikation gewährleistet und weiter optimiert werden.

Bezüglich der Zeitkonstante des inneren Regelkreises ist zu untersuchen, inwiefern die Integration des a priori Wissen über die geplanten Trajektorien der Nachbarbalken in den Regelalgorithmus die Performanz optimiert. Dies kann zusätzlich durch den direkten Informationsaustausch zwischen den STM-Knoten über ihre aktuelle Position noch ergänzt werden und erfordert gegebenenfalls eine Anpassung der Übertragungsprotokolle inklusive der Einrichtung von priorisierten Kommunikationswegen.

# Literaturverzeichnis

- [1] ABEL, Bollig A. D.: *Rapid control prototyping : Methoden und Anwendungen*. Berlin ; Heidelberg : Springer, 2013. – ISBN 978–3–642–31944–0
- [2] ABEL, D.: *Regelungstechnik*. Aachener Forschungsgesellschaft Regelungstechnik e.V., 2013 (37. Auflage). – ISBN 3–8107–0067–3
- [3] AHN, Hyo-Sung ; CHEN, Yang Q. ; MOORE, K.L.: Iterative Learning Control: Brief Survey and Categorization. In: *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on* 37 (2007), Nov, Nr. 6, S. 1099–1121. <http://dx.doi.org/10.1109/TSMCC.2007.905759>. – DOI 10.1109/TSMCC.2007.905759. – ISSN 1094–6977
- [4] AKYILDIZ, I. F. ; SU, W. ; SANKARASUBRAMANIAM, Y. ; CAYIRCI, E.: Wireless sensor networks: a survey. In: *Computer Networks* 38 (2002), S. 393–422
- [5] ALLIED TELESIS: *AT-FS708LE Datasheet, Unmanaged 8 port fast ethernet switch featuring low power technology*
- [6] ALSAPROJECT: *Advanced Linux Sound Architecture ALSA*. – [05.01.2016]  
<http://www.alsa-project.org>
- [7] ARIMOTO, Suguru ; KAWAMURA, Sadao ; MIYAZAKI, Fumio: Bettering operation of Robots by learning. In: *Journal of Robotic Systems* 1 (1984), Nr. 2, 123–140. <http://dx.doi.org/10.1002/rob.4620010203>. – DOI 10.1002/rob.4620010203. – ISSN 1097–4563
- [8] In: BALAGEAS, Daniel: *Introduction to Structural Health Monitoring*. ISTE, 2010. – ISBN 9780470612071, 13–43
- [9] BANERJEE, S. ; SETHIA, D. ; MITTAL, T. ; ARORA, U. ; CHAUHAN, A.: Secure sensor node with Raspberry Pi. In: *Multimedia, Signal Processing and Communication Technologies (IMPACT), 2013 International Conference on*, 2013, S. 26–30

- [10] BRADEN, R: *RFC 1122: Requirements for Internet Hosts - Communication Layers*, 1989. <http://tools.ietf.org/html/rfc1122>
- [11] BRADEN, R: *RFC 1123: Requirements for Internet Hosts - Application and Support*, 1989. <http://tools.ietf.org/html/rfc1123>
- [12] BRAHIMI, Belynda ; RONDEAU, Eric ; AUBRUN, Christophe: Comparison between Networked Control System behaviour based on CAN and Switched Ethernet networks. In: *CoRR abs/cs/0611149* (2006)
- [13] BRINDHA, M. ; MENDIRATTA, J. K.: Networked Control System – A Survey. In: *International Journal of Modern Education and Computer Science (IJMECS)* Vol.5 (2013), Nr. 6, S. 42–48
- [14] BRISTOW, D.A. ; THARAYIL, M. ; ALLEYNE, A.G.: A survey of iterative learning control. In: *Control Systems, IEEE* 26 (2006), June, Nr. 3, S. 96–114. <http://dx.doi.org/10.1109/MCS.2006.1636313>. – DOI 10.1109/MCS.2006.1636313. – ISSN 1066–033X
- [15] CATTAFESTA III, Louis N. ; SHEPLAK, Mark: Actuators for Active Flow Control. In: *Annual Review of Fluid Mechanics* 43 (2011), Nr. 1, 247–272. <http://dx.doi.org/10.1146/annurev-fluid-122109-160634>. – DOI 10.1146/annurev-fluid-122109-160634
- [16] CERVIN, Anton ; HENRIKSSON, Dan ; LINCOLN, Bo ; EKER, Johan ; ÅRZÉN, Karl-Erik: How Does Control Timing Affect Performance? Analysis and Simulation of Timing Using Jitterbug and TrueTime. In: *IEEE Control Systems Magazine* 23 (2003), Juni, Nr. 3, S. 16–30
- [17] CERVIN, Anton ; HENRIKSSON, Dan ; OHLIN, Martin: *TrueTime 2.0 beta – Reference Manual*. Department of Automatic Control, Lund University, 2010
- [18] CERVIN, Anton ; OHLIN, Martin ; HENRIKSSON, Dan: Simulation of Networked Control Systems Using TrueTime. In: *Proc. 3rd International Workshop on Networked Control Systems: Tolerant to Faults*. Nancy, France, Juni 2007
- [19] CHEN, Chih-Keng ; HWANG, James: PD-type iterative learning control for trajectory tracking of a pneumatic X-Y table with disturbances. In: *Robotics and Automation, 2004. Proceedings. ICRA '04. 2004 IEEE International Conference on* Bd. 4, 2004. – ISSN 1050–4729, S. 3500–3505 Vol.4

- [20] CHEN, Yangquan ; GONG, Zhiming ; WEN, Changyun: Analysis of a high-order iterative learning control algorithm for uncertain nonlinear systems with state delays\*. In: *Automatica* 34 (1998), Nr. 3, 345 - 353. [http://dx.doi.org/http://dx.doi.org/10.1016/S0005-1098\(97\)00196-9](http://dx.doi.org/http://dx.doi.org/10.1016/S0005-1098(97)00196-9). – DOI [http://dx.doi.org/10.1016/S0005-1098\(97\)00196-9](http://dx.doi.org/10.1016/S0005-1098(97)00196-9). – ISSN 0005-1098
- [21] COTA, O. F. ; SCHLÖSSER, M. ; SCHIEK, M. ; STIEGLITZ, T. ; GIERTH-MUEHLEN, M. ; PLACHTA, D.: iNODE in-vivo testing for selective vagus nerve recording and stimulation. In: *Neural Engineering (NER), 2015 7th International IEEE/EMBS Conference on*, 2015, S. 525-528
- [22] CREATIVE LABS: *Sound Blaster Play!*. – [05.01.2016]  
<http://de.creative.com/p/sound-blaster/sound-blaster-play>
- [23] DELEUZE, Christophe: Content networks. In: *The Internet Protocol Journal* 7 (2004), Nr. 2, S. 2-11
- [24] DRATH, R. ; HORCH, A.: Industrie 4.0: Hit or Hype? [Industry Forum]. In: *IEEE Industrial Electronics Magazine* 8 (2014), June, Nr. 2, S. 56-58. – ISSN 1932-4529
- [25] DUECK, M. ; ABEL, D. ; WAASEN, S. van ; SCHIEK, M.: Transversal Surface Wave Control by Gain switching iterative Learning improving Research on active turbulent Flow Control. In: *Control Conference (AUCC), 2015 5th Australian*, 2015
- [26] DUECK, M. ; KAPARAKI, M. ; SRIVASTAVA, S. ; WAASEN, S. van ; SCHIEK, M.: Development of a real time actuation control in a network-simulation framework for active drag reduction in turbulent flow. In: *Automatic Control Conference (CACs), 2013 CACS International*, 2013, S. 256-261
- [27] DUECK, M. ; SCHLOESSER, M. ; KAPARAKI, M. ; SRIVASTAVA, S. ; WAASEN, S. van ; SCHIEK, M.: Raspberry Pi based testbed verifying TrueTime network model parameters for application in distributed active turbulent flow control. In: *SICE Annual Conference (SICE), 2014 Proceedings of the*, 2014, S. 1970-1975
- [28] DUECK, M. ; SCHLOESSER, M. ; WAASEN, S. van ; SCHIEK, M.: Deterministic Transport Protocol Verified by a Real-Time Actuator and Sensor Network Simulation for Distributed Active Turbulent Flow Control. Version: September 2015. [http://dx.doi.org/10.1007/978-81-322-2529-4\\_3](http://dx.doi.org/10.1007/978-81-322-2529-4_3). In: NAGAR, Atulya (Hrsg.) ; MOHAPATRA, Durga P. (Hrsg.) ; CHAKI, Nabendu (Hrsg.): *Proceedings of 3rd International*

- Conference on Advanced Computing, Networking and Informatics* Bd. 44. Springer India, September 2015. – DOI 10.1007/978-81-322-2529-4\_3. – ISBN 978-81-322-2528-7, 29-38
- [29] DUECK, M. ; SCHLOESSER, M. ; WAASEN, S. van ; SCHIEK, M.: Ethernet based time synchronization for Raspberry Pi network improving network model verification for distributed active turbulent flow control. In: *Control Theory and Technology* 13 (2015), May, Nr. 2, S. 89–95. <http://dx.doi.org/10.1007/s11768-015-4143-1>. – DOI 10.1007/s11768-015-4143-1
- [30] DUECK, M. ; TRABERT, J. ; SEIDLER, F. ; SILEX, W. ; WAASEN, S. van ; SCHIEK, M. ; ABEL, D. ; B. CASTELAN, E.: Control of Nonlinear Coupled Electromagnetic Actuators for Active Drag Reduction in Turbulent Flow. In: *Automation 2016, Secure & reliable in the digital world, VDI-Berichte 2284, 07.-08. Juni 2016, Baden-Baden*, VDI Verlag, 2016. – ISBN 978-3-18-092284-0, S. 873–884
- [31] DUECK, M. ; VOELKEL, S. ; WAASEN, S. van ; SCHIEK, M. ; ABEL, D.: Entwicklung einer echtzeitigen Aktuator-Ansteuerung mit Transienten-Glättung in LabVIEW Real-Time zur Strömungsregelung durch transversale Oberflächenwellen. In: *Automation 2015, Benefits of Change - the Future of Automation, VDI-Berichte 2258, 11.-12. Juni 2015, Baden-Baden*, VDI Verlag, 2015. – ISBN 978-3-18-092258-4, S. 203–212
- [32] FELSER, M.: *PROFIBUS Handbuch – Eine Sammlung von Erläuterungen zu PROFIBUS Netzwerken*. Galileo Computing, 2015 (1. Auflage). – ISBN 978-3.7375-5470-1
- [33] GROSSE, Sebastian ; SCHRÖDER, Wolfgang: Dynamic wall-shear stress measurements in turbulent pipe flow using the micro-pillar sensor MPS3. In: *International Journal of Heat and Fluid Flow* 29 (2008), Nr. 3, 830 - 840. <http://dx.doi.org/http://dx.doi.org/10.1016/j.ijheatfluidflow.2008.01.008>. – DOI <http://dx.doi.org/10.1016/j.ijheatfluidflow.2008.01.008>. – ISSN 0142-727X. – The Fifth International Symposium on Turbulence and Shear Flow Phenomena (TSFP5)
- [34] GUMUDAVELLI, S. ; GURKAN, D. ; WANG, R.: Emulated network of IEEE 1451 application with multiple smart sensor reports. In: *Sensors Applications Symposium, 2009. SAS 2009. IEEE*, 2009, S. 304–308
- [35] HAKENBERG, Mathias ; ABEL, Dirk: Frequency-Selective Galerkin Models for Turbulent Boundary Layer Drag Reduction.

- (2014), 2014-08-24 - 2014-08-29, 3979-3984. <http://dx.doi.org/10.3182/20140824-6-ZA-1003.02743>. – DOI 10.3182/20140824-6-ZA-1003.02743
- [36] HE, Zhonghai ; ZHAO, Yuqian: Modeling in Respiratory Movement Using LabVIEW and Simulink. Version: 2011. <http://dx.doi.org/10.5772/13134>. In: *Modeling, Programming and Simulations Using LabVIEW Software*, Dr. Ricardo De Asmundis (Ed.). InTech, 2011. – DOI 10.5772/13134. – ISBN 978-953-307-521-1, 137-160
- [37] INSTITUTE OF ELECTRICAL AND ELECTRONICS ENGINEERS, INC.: *IEEE Std 802-1990: IEEE Standards for Local and Metropolitan Networks: Overview and Architecture New York:1990*. 1990. – 3 Park Avenue, New York, NY 10016-5997, USA
- [38] INSTITUTE OF ELECTRICAL AND ELECTRONICS ENGINEERS, INC.: *IEEE Std 1451.1-1999, Standard for a Smart Transducer Interface for Sensors and Actuators – Network Capable Application Processor (NCAP) Information Model*. June 1999. – 3 Park Avenue, New York, NY 10016-5997, USA
- [39] INSTITUTE OF ELECTRICAL AND ELECTRONICS ENGINEERS, INC.: *IEEE Std 1588-2008, Standard for a Precision Clock Synchronization Protocol for Networked Measurement and Control Systems*. 2008. – 3 Park Avenue, New York, NY 10016-5997, USA
- [40] INTERNATIONAL ELECTROTECHNICAL COMMISSION: *Industrial communication networks - Profiles*. – IEC 61784-1
- [41] INTERNATIONAL ORGANIZATION FOR STANDARDIZATION: *Road vehicles – Controller area network (CAN)*. – ISO 11898
- [42] INTERNATIONAL ORGANIZATION FOR STANDARDIZATION: *Road vehicles – FlexRay communications system*. – ISO 17458
- [43] INTERNATIONAL ORGANIZATION FOR STANDARDIZATION, INTERNATIONAL ELECTROTECHNICAL COMMISSION: *Information technology - Open Systems Interconnection - Basic Reference Model: The Basic Model*. 1994. – ISO/IEC 7498-1:1994(E)
- [44] INTERNET ENGINEERING TASK FORCE (Hrsg.): *RFC 791 Internet Protocol - DARPA Internet Programm, Protocol Specification*. Internet Engineering Task Force, September 1981. <http://tools.ietf.org/html/rfc791>

- [45] IONETE, C. ; SENDRESCU, D. ; POPESCU, D. ; IONETE, A.: Ethernet for networked control an experimental test bench. In: *SICE Annual Conference 2010, Proceedings of*, 2010, S. 3556–3559
- [46] KARNIADAKIS, G.E. ; CHOI, Kwing-So: MECHANISMS ON TRANSVERSE MOTIONS IN TURBULENT WALL FLOWS. In: *Annual Review of Fluid Mechanics* 35 (2003), Nr. 1, 45-62. <http://dx.doi.org/10.1146/annurev.fluid.35.101101.161213>. – DOI 10.1146/annurev.fluid.35.101101.161213
- [47] KERSKEN, S.: *IT-Handbuch für Fachinformatiker*. Galileo Computing, 2009 (4. Auflage). – ISBN 978-3-8362-1420-9
- [48] KEYENCE: *LK-H022 Sensor Head Spot Type, Laser Class 2*. – [10.06.2015], <http://www.keyence.com/products/measure/laser-1d/lk-g5000/models/lk-h022/index.jsp>
- [49] KLUMPP, Stephan ; MEINKE, Matthias ; SCHRÖDER, Wolfgang: Friction Drag Variation via Spanwise Transversal Surface Waves. In: *Flow, Turbulence and Combustion* 87 (2011), Nr. 1, 33–53. <http://dx.doi.org/10.1007/s10494-011-9326-3>. – DOI 10.1007/s10494-011-9326-3. – ISSN 1573-1987
- [50] KOH, Seong R. ; MEYSONNAT, Pascal ; MEINKE, Matthias ; SCHRÖDER, Wolfgang: Drag Reduction Via Spanwise Transversal Surface Waves at High Reynolds Numbers. In: *Flow, Turbulence and Combustion* 95 (2015), Nr. 1, 169-190. <http://dx.doi.org/10.1007/s10494-015-9614-4>. – DOI 10.1007/s10494-015-9614-4. – ISSN 1386-6184
- [51] LEHRBAUM, R.: Using the PC architecture in embedded applications. In: *WESCON/'93. Conference Record*, 1993. – ISSN 1095-791X, S. 610–620
- [52] LI, W. ; JESSEN, W. ; ROGGENKAMP, D. ; KLAAS, M. ; SILEX, W. ; SCHIEK, M. ; SCHROEDER, W.: Turbulent drag reduction by spanwise traveling ribbed surface waves. In: *European Journal of Mechanics - B/Fluids* (2015). <http://dx.doi.org/http://dx.doi.org/10.1016/j.euromechflu.2015.03.009>. – DOI <http://dx.doi.org/10.1016/j.euromechflu.2015.03.009>. – ISSN 0997-7546
- [53] LONGMAN, Richard: On the interaction between theory experiments and simulation in developing practical learning control algorithms. In: *International Journal of Applied Mathematics and Computer Science* 13 (2003), Nr. 1, 101-111. <http://eudml.org/doc/207618>

- [54] LONGMAN, Richard W.: Iterative learning control and repetitive control for engineering practice. In: *International Journal of Control* 73 (2000), Nr. 10, 930-954. <http://dx.doi.org/10.1080/002071700405905>. – DOI 10.1080/002071700405905
- [55] MEYSONNAT, Pascal S. ; ROGGENKAMP, Dorothee ; LI, Wenfeng ; ROIDL, Benedikt ; SCHRÖDER, Wolfgang: Experimental and numerical investigation of transversal traveling surface waves for drag reduction. In: *European Journal of Mechanics - B/Fluids* 55, Part 2 (2016), 313 - 323. <http://dx.doi.org/http://dx.doi.org/10.1016/j.euromechflu.2015.07.001>. – DOI <http://dx.doi.org/10.1016/j.euromechflu.2015.07.001>. – ISSN 0997-7546. – Vortical Structures and Wall Turbulence
- [56] MILLS, Dave L.: *Internet time synchronization: The Network Time Protocol*. Network Working Group Request for Comments: 1129, Oktober 1989
- [57] MODMYPI: *Installing the RasClock – Raspberry Pi Real Time Clock*. – [02.01.2016]  
<http://www.raspberrypi.org/faqs>
- [58] MOORE, Kevin L.: A non-standard iterative learning control approach to tracking periodic signals in discrete-time non-linear systems. In: *International Journal of Control* 73 (2000), Nr. 10, 955-967. <http://dx.doi.org/10.1080/002071700405914>. – DOI 10.1080/002071700405914
- [59] MORITZ, G. ; PRUTER, S. ; TIMMERMAN, D. ; GOLATOWSKI, F.: Real-time service-oriented communication protocols on resource constrained devices. In: *Computer Science and Information Technology, 2008. IMC-SIT 2008. International Multiconference on*, 2008, S. 695-701
- [60] NATIONAL INSTRUMENTS: *Configuring the Data Transfer Request Condition Property in NI-DAQmx*. – [17.11.2015],  
<http://www.ni.com/white-paper/4402/en/>
- [61] NATIONAL INSTRUMENTS: *Products and Services*. – [17.11.2015],  
<http://www.ni.com/products/>
- [62] NATIONAL INSTRUMENTS CORPORATION: *National Instruments: Test, Measurement, and Embedded Systems*. – [01.12.2015],  
<http://www.ni.com>

- [63] NETWORK TIME FOUNDATION: *The Network Time Protocol*. – [22.10.2014], <http://www.ntp.org/>
- [64] NEUMANN, Peter: Communication in industrial automation – What is going on? In: *Control Engineering Practice* 15 (2007), Nr. 11, 1332 - 1347. <http://dx.doi.org/http://dx.doi.org/10.1016/j.conengprac.2006.10.004>. – DOI <http://dx.doi.org/10.1016/j.conengprac.2006.10.004>. – ISSN 0967–0661. – Special Issue on Manufacturing Plant Control: Challenges and Issues INCOM 2004 11th {IFAC} INCOM'04 Symposium on Information Control Problems in Manufacturing
- [65] NOLTE, T. ; HANSSON, H. ; BELLO, L.L.: Automotive communications-past, current and future. In: *Emerging Technologies and Factory Automation, 2005. ETFA 2005. 10th IEEE Conference on* Bd. 1, 2005, S. 8 pp.–992
- [66] NORRLÖF, Mikael ; GUNNARSSON, Svante: Time and frequency domain convergence properties in iterative learning control. In: *International Journal of Control* 75 (2002), Nr. 14, 1114-1126. <http://dx.doi.org/10.1080/00207170210159122>. – DOI 10.1080/00207170210159122
- [67] POSTEL, J.: User Datagram Protocol. In: *Request For Comment 768* (1980)
- [68] POSTEL, Jon B.: *Transmission Control Protocol*. Internet RFC 793, September 1981
- [69] PYTA, Lorenz ; ABEL, Dirk: Model based control of the incompressible Navier-Stokes-Equations using interpolatory model reduction. In: *Decision and Control (CDC), 2015 IEEE 54th Annual Conference on*, 2015, S. 7316–7321
- [70] RASPBERRY PI FOUNDATION: *Raspberry Pi FAQs*. – [10.02.2014] <http://www.raspberrypi.org/faqs>
- [71] ROCKEL, Hellmar ; KONIGORSKI, Ulrich: Iterativ Lernende Regelungen, Teil I: Ein parametrischer Rahmen (A General Framework for Parametric Iterative Learning Control). In: *Automatisierungstechnik* 55 (2007), Nr. 3, S. 119–126
- [72] ROGGENKAMP, D ; LI, W. ; MEYSONNAT, P. ; KLAAS, M. ; SCHRÖDER, W.: Experimental analysis of turbulent friction drag reduction due to

- spanwise transversal surface waves. In: *Ninth International Symposium on Turbulence and Shear Flow Phenomena*, 2015
- [73] SCHIEK, M. ; SILEX, W. ; DUECK, M ; WAASEN, S. van: Flexible electromagnetic actuating system for research on turbulence flow control. In: *Proceedings 106. SEI-Tagung 2015*, 2015
- [74] SCHLOESSER, M. ; COTA, O. ; HEIL, R. ; BRUSIUS, J. ; OFFENHAUSSER, A. ; WAASEN, S.V. ; SCHIEK, M.: Embedded device for simultaneous recording and stimulation for retina implant research. In: *SENSORS, 2013 IEEE*, 2013. – ISSN 1930–0395, S. 1–4
- [75] SCHLOESSER, Mario ; SCHNITZER, Andreas ; YING, Hong ; SILEX, Carmen ; SCHIEK, Michael: iNODE: intelligent Network Operating Device for Neurological and Neurophysiological Research. Version: 2009. [http://dx.doi.org/10.1007/978-3-540-89208-3\\_407](http://dx.doi.org/10.1007/978-3-540-89208-3_407). In: VANDER SLOTEN, Jos (Hrsg.) ; VERDONCK, Pascal (Hrsg.) ; NYSSSEN, Marc (Hrsg.) ; HAU-EISEN, Jens (Hrsg.): *4th European Conference of the International Federation for Medical and Biological Engineering* Bd. 22. Springer Berlin Heidelberg, 2009. – DOI 10.1007/978-3-540-89208-3\_407. – ISBN 978-3-540-89207-6, 1710-1713
- [76] SCHMIDT, Douglas C.: *The ADAPTIVE Communication Environment (ACE)*. – [05.01.2016], <http://www.cs.wustl.edu/~schmidt/ACE.html>
- [77] SCHULZRRINNE, H. ; CASNER, S. ; FREDERICK, R. ; JACOBSON, V.: RFC 3550: RTP: A Transport Protocol for Real-Time Applications / IETF. Version: 2003. [www.ietf.org/rfc/rfc3550.txt](http://www.ietf.org/rfc/rfc3550.txt). 2003. – Forschungsbericht
- [78] SMITH, D. ; OLARIU, C. ; PERRY, P. ; MURPHY, J.: Impact of non-deterministic software execution times in SmartGrid applications. In: *Signals and Systems Conference (ISSC), 2015 26th Irish*, 2015, S. 1–5
- [79] SONG, E.Y. ; LEE, K.: Understanding IEEE 1451-Networked smart transducer interface standard - What is a smart transducer? In: *Instrumentation Measurement Magazine, IEEE* 11 (2008), April, Nr. 2, S. 11–17. <http://dx.doi.org/10.1109/MIM.2008.4483728>. – DOI 10.1109/MIM.2008.4483728. – ISSN 1094–6969
- [80] SPEEDGOAT GMBH: *Speedgoat - x86 / FPGA real-time hardware for Simulink*. – [01.12.2015], <http://www.speedgoat.ch>

- [81] TAMANO, Shinji ; ITOH, Motoyuki: Drag reduction in turbulent boundary layers by spanwise traveling waves with wall deformation. In: *Journal of Turbulence* 13 (2012), N9. <http://dx.doi.org/10.1080/14685248.2012.655743>. – DOI 10.1080/14685248.2012.655743
- [82] TEKIN, R.: MATLAB and LabVIEW in modeling, analysis and real time control of a motion control system. In: *Control and Automation (ICCA), 2010 8th IEEE International Conference on*, 2010. – ISSN 1948–3449, S. 2077–2081
- [83] THOMANN: *the t.amp TSA 4-1300*. – [27.02.2016]  
[http://www.thomann.de/de/the\\_tamp\\_tsa\\_4\\_1300.htm](http://www.thomann.de/de/the_tamp_tsa_4_1300.htm)
- [84] TOBAGI, F.A. ; DALGIC, I.: Performance evaluation of 10Base-T and 100Base-T Ethernet carrying multimedia traffic. In: *Selected Areas in Communications, IEEE Journal on* 14 (1996), Nr. 7, S. 1436–1454. <http://dx.doi.org/10.1109/49.536490>. – DOI 10.1109/49.536490. – ISSN 0733–8716
- [85] VERÍSSIMO, Paulo: Distributed Systems (2Nd Ed.). Version: 1993. <http://dl.acm.org/citation.cfm?id=302430.302447>. New York, NY, USA : ACM Press/Addison-Wesley Publishing Co., 1993. – ISBN 0–201–62427–3, Kapitel Real-time Communication, 447–490
- [86] VIEGAS, V. ; PEREIRA, M. ; GIRAO, P.: A brief tutorial on the IEEE 1451.1 Standard - Part 13 in a series of tutorials in instrumentation and measurement. In: *Instrumentation Measurement Magazine, IEEE* 11 (2008), April, Nr. 2, S. 38–46. <http://dx.doi.org/10.1109/MIM.2008.4483732>. – DOI 10.1109/MIM.2008.4483732. – ISSN 1094–6969
- [87] VOELKEL, S. ; DUECK, M. ; SCHLOESSER, M ; WAASEN, S. van ; SCHIEK, M.: Entwicklung einer echtzeitigen Aktuator-Ansteuerung mit Transienten-Glättung in LabVIEW Real-Time. In: RAHMAN JAMAL, Roland H. (Hrsg.): *Virtuelle Instrumente in der Praxis* Bd. Begleitband zum 20. VIP Kongress. VDE Verlag, October 2015. – ISBN 978–3–8007–3669–0, S. 222–226
- [88] WANG, Wei ; NESIC, D. ; POSTOYAN, R.: Emulated controller design for networked control systems implemented on FlexRay. In: *Decision and Control (CDC), 2014 IEEE 53rd Annual Conference on*, 2014, S. 5272–5277
- [89] WANG, Youqing ; GAO, Furong ; III, Francis J. D.: Survey on iterative learning control, repetitive control, and run-to-run control. In: *Journal of Process Control* 19 (2009), Nr. 10, 1589 - 1600. <http://>

- [dx.doi.org/http://dx.doi.org/10.1016/j.jprocont.2009.09.006](http://dx.doi.org/http://dx.doi.org/10.1016/j.jprocont.2009.09.006). – DOI <http://dx.doi.org/10.1016/j.jprocont.2009.09.006>. – ISSN 0959–1524
- [90] WILSON, David A. ; WOODS, Stan ; SONG, Yuyin: *Open1451 Gaithersburg – An Open Implementation of IEEE 1451*, MONTH = June, YEAR = 2003
- [91] YANG, T.-C.: Networked control system: a brief survey. In: *Control Theory and Applications, IEE Proceedings - 153* (2006), July, Nr. 4, S. 403–412. <http://dx.doi.org/10.1049/ip-cta:20050178>. – DOI 10.1049/ip-cta:20050178. – ISSN 1350–2379
- [92] YING, H. ; SCHLÖSSER, M. ; SCHNITZER, A. ; SCHÄFER, T. ; SCHLÄFKE, M. E. ; LEONHARDT, S. ; SCHIEK, M.: Distributed Intelligent Sensor Network for the Rehabilitation of Parkinson-Patients. In: *IEEE Transactions on Information Technology in Biomedicine* 15 (2011), March, Nr. 2, S. 268–276. <http://dx.doi.org/10.1109/TITB.2010.2095463>. – DOI 10.1109/TITB.2010.2095463. – ISSN 1089–7771
- [93] ZIMMERMANN, H.: OSI Reference Model–The ISO Model of Architecture for Open Systems Interconnection. In: *Communications, IEEE Transactions on* 28 (1980), Apr, Nr. 4, S. 425–432. <http://dx.doi.org/10.1109/TCOM.1980.1094702>. – DOI 10.1109/TCOM.1980.1094702. – ISSN 0090–6778



Band / Volume 336

**Production and Characterization of Monodisperse Uranium Particles for Nuclear Safeguards Applications**

A. Knott (2016), A-H, 123, xxviii, xiii pp

ISBN: 978-3-95806-172-9

Band / Volume 337

**3D hydrological simulation of a forested headwater catchment: Spatio-temporal validation and scale dependent parameterization**

Z. Fang (2016), XVII, 119 pp

ISBN: 978-3-95806-174-3

Band / Volume 338

**Influence of Thermomechanical Treatment on High Temperature Properties of Laves Phase Strengthened Ferritic Steels**

M. Talík (2016), xxiii, 130 pp

ISBN: 978-3-95806-175-0

Band / Volume 339

**Groundwater recharge in Slovenia  
Results of a bilateral German-Slovenian Research project**

Mišo Andjelov, Zlatko Mikulič, Björn Tetzlaff, Jože Uhan & Frank Wendland (2016)

ISBN: 978-3-95806-177-4

Band / Volume 340

**Atomic oxygen derived from SCIAMACHY O(<sup>1</sup>S)  
and OH airglow measurements in the Mesopause region**

Y. Zhu (2016), 206 pp

ISBN: 978-3-95806-178-1

Band / Volume 341

**Diagnostic Mirror Concept Development for Use  
in the Complex Environment of a Fusion Reactor**

A. Krimmer (2016), x, 123 pp

ISBN: 978-3-95806-180-4

Band / Volume 342

**Long-Term Measurements of the Radiation Exposure of the Inhabitants  
of Radioactively Contaminated Regions of Belarus –  
The Korma Report II (1998 – 2015)**

P. Zoriy, H. Dederichs, J. Pillath, B. Heuel-Fabianek, P. Hill, R. Lennartz (2016), ca 105 pp

ISBN: 978-3-95806-181-1

Band / Volume 343

**Entwicklung und Charakterisierung von Nickel/Gadolinium-Ceroxid-basierten Anoden für die metallgestützte Festoxid-Brennstoffzelle**

V. Rojek-Wöckner (2016), XVI, 136 pp

ISBN: 978-3-95806-182-8

Band / Volume 344

**Reaction-diffusion modelling of hydrogen retention and release mechanisms in beryllium**

M. Wensing (2016), 100 pp

ISBN: 978-3-95806-184-2

Band / Volume 345

**Light Management by Intermediate Reflectors in Silicon-based Tandem Solar Cells**

A. Hoffmann (2016), 199 pp

ISBN: 978-3-95806-186-6

Band / Volume 346

**Design of a Highly Efficient Solid Oxide Fuel Cell System with Integrated Safety Gas Generation**

M. Engelbracht (2016), 190 pp

ISBN: 978-3-95806-189-7

Band / Volume 347

**On model and measurement uncertainty in predicting land surface carbon fluxes**

H. Post (2016), xviii, 135 pp

ISBN: 978-3-95806-190-3

Band / Volume 348

**Bipolarplattenmaterialien für Polymer-Elektrolyt-Membran Elektrolyse**

M. Langemann (2016), I-III, 189, IV-XVIII pp

ISBN: 978-3-95806-192-7

Band / Volume 349

**Modellbasierte Ansteuerung räumlich ausgedehnter Aktuator- und Sensornetzwerke in der Strömungsregelung**

M. Dück (2016), XIII, 153 pp

ISBN: 978-3-95806-193-4

Weitere **Schriften des Verlags im Forschungszentrum Jülich** unter  
<http://wwwzb1.fz-juelich.de/verlagextern1/index.asp>



**Energie & Umwelt /  
Energy & Environment  
Band / Volume 349  
ISBN 978-3-95806-193-4**

