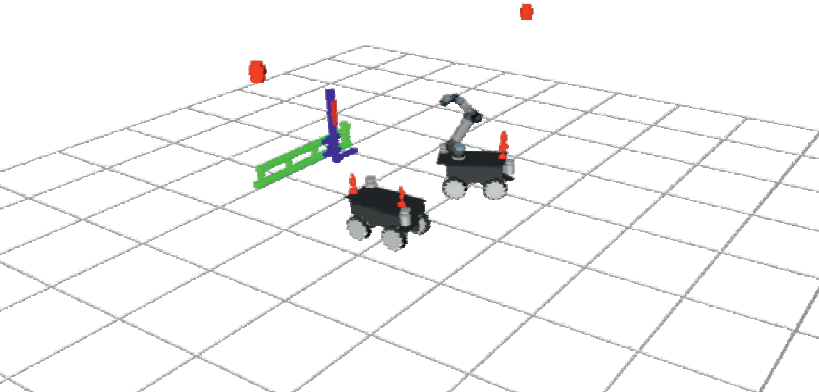


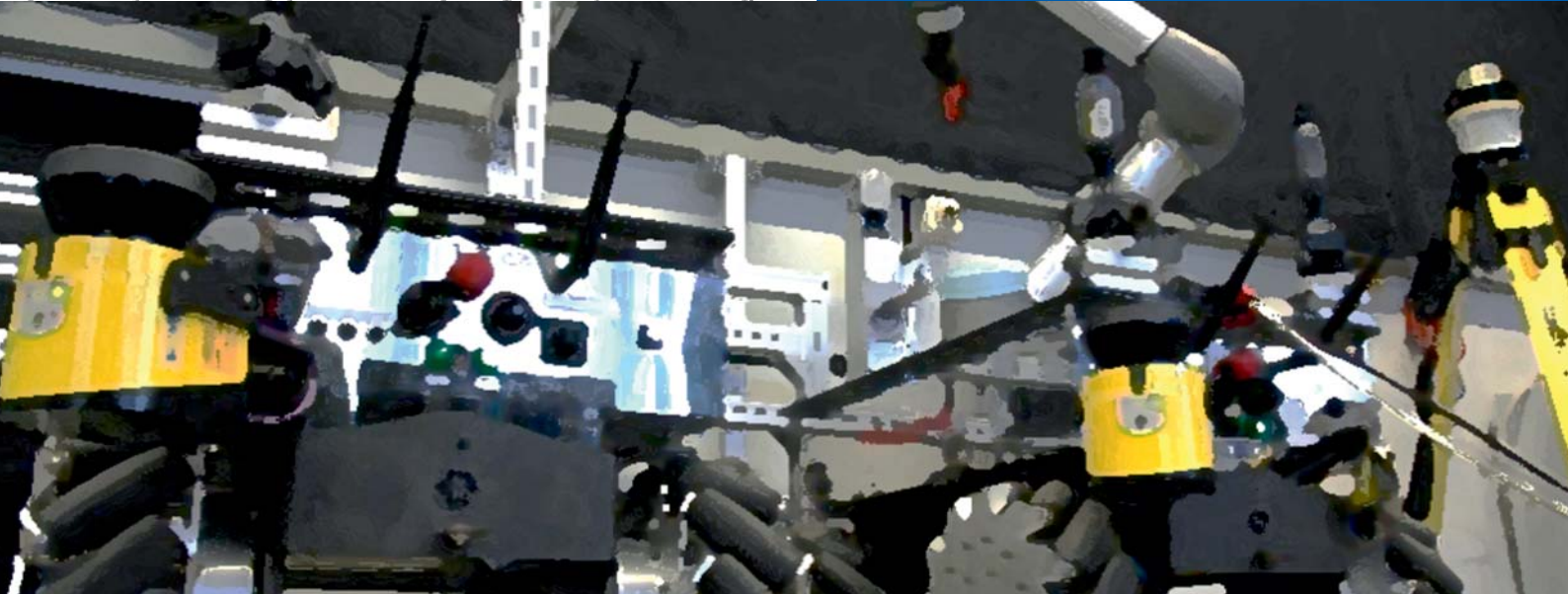


5. IFToMM D-A-CH Konferenz 2019

26./27. Februar 2019, RWTH Aachen



- Biomechanische Systeme
- Kinematische Analyse und Synthese
- Getriebetechnik
- Regelung von Mechanismen und Maschinen
- Mikromechanische Systeme
- Mehrkörpersysteme
- Sicherheit und Zuverlässigkeit
- Robotik und Mechatronik
- Rotordynamik
- Nachhaltigkeit, Ressourcen- und Energieeffizienz
- Fahrzeug- und Transportsysteme
- Tribologie
- Schwingungstechnik und Maschinendynamik
- Nachgiebige Mechanismen



IFToMM D-A-CH Konferenz 2019

IFToMM Member Organizations
Austria, Germany, Switzerland

26.-27. Februar 2019, RWTH Aachen University

TAGUNGSBAND



Local Organization:

- Mathias Hüsing, RWTH Aachen
- Claudia Cornely, RWTH Aachen

Program Committee:

- Maik Berger, TU Chemnitz
- Torsten Bertram, TU Dortmund
- Hannes Bleuler, EPFL
- Tobias Bruckmann, Uni Duisburg-Essen
- Burkhard Corves, RWTH Aachen
- Francisco Geu, Uni Duisburg-Essen
- Mathias Hüsing, RWTH Aachen
- Manfred Husty, Uni Innsbruck
- Andrés Kecskeméthy, Uni Duisburg-Essen
- Jörn Malzahn, IIT Genova
- Malte Oeljeklaus, TU Dortmund
- Jürg Schiffmann, EPFL

Technical Chair

- Jörn Malzahn, IIT Genova

Publications Chair

- Tobias Bruckmann, Uni Duisburg-Essen

Publicity Chair

- Mathias Hüsing, RWTH Aachen

Impressum

ISBN 978-3-940402-21-9

DOI 10.17185/dupublico/47897

Herausgeber

Universität Duisburg-Essen

Erscheinungsort

DuEPublico, Duisburg-Essen Publications Online

Universität Duisburg-Essen

<http://dupublico.uni-duisburg-essen.de>

© 2019



5. IFToMM D-A-CH Konferenz 2019

26./27. Februar 2019, RWTH Aachen

Programm 26. Februar

Ab 8:00 Uhr: Registrierung

Ab 8:20 Uhr: Begrüßung durch Burkhard Corves

8:30 – 10:30 Uhr: Session I: Kinematik (Leitung: Stefan Heinrich)

Claudio Villegas (RWTH Aachen), Mathias Hüsing und Burkhard Corves

Synthesis of Mechanisms Based on their Joint-Forces

Alexandra Mercader (TU München), Anian Leyerer, Franz Irlinger und Tim Lueth

Deep Learning zur Viergelenksynthese unter Verwendung der Fourier-Koeffizienten der Koppelkurve

Thorsten Speicher (TU Chemnitz) und Maik Berger

Methodik zur Auswahl und Synthese von Bewegungsgesetzen für mechatronische Systeme zur effizienten Prozesssteigerung

Liang Chen (TU Chemnitz), Maik Berger und Fan Yin

Design und Konstruktion eines neuen Konzepts für Mehrteilgreifer

10:30 – 10:45 Uhr: Kaffeepause

10:45 – 12:45 Uhr: Session II: Robotik (Leitung: Martin Pfurner)

Michael Gerbl (Univ. Innsbruck), Michael Pieber und Johannes Gerstmayr

Reconfiguration of Modular Robots with triangular structure

Veronika Zumpe (KUKA Deutschland), Johannes Horeis, Mathias Hüsing und Burkhard Corves

Pre-Processing für taskbasierte Kinematik-Synthese

Justus Kopp (Bosch Rexroth AG, Priv. Univ. UMIT, Hall/Tirol), Bertold Bader und Frank Woittennek

Online-Optimierung von Geschwindigkeitsprofilen entlang vorgegebener Bahnen unter Berücksichtigung von Achsruckbeschränkungen

Martin Sereinig (Univ. Innsbruck), Johannes Gerstmayr und Martin Pfurner

Kollisionsvermeidung und Pfadplanung im diskreten Gelenkwinkelraum redundanter Manipulatoren



5. IFToMM D-A-CH Konferenz 2019

26./27. Februar 2019, RWTH Aachen

12:45 – 13:45 Uhr: Mittagessen

14:00 – 16:00 Uhr: Session III: Robotik (Leitung: Mathias Hüsing)

Amirreza Shahidi (RWTH Aachen), Tim Detert, Mathias Hüsing und Burkhard Corves

Agile Montagesystem durch Konsens von Roboter- und Sensornetzwerk – Eine Einführung zur Industrie 4.0

Philipp Wabnitz (TU Chemnitz), Andreas Heine und Maik Berger

Vergleich von haptischen Steuerungsvarianten zur manuellen Bedienung von Robotersystemen

Tobias Bruckmann (Univ. Duisburg-Essen) und Arnim Spengler

Simulation automatisierter Bauprozesse unter Einsatz von Seilrobotern

Bertold Bongardt (TU Braunschweig)

Eine analytische Lösung der inversen Kinematik allgemeiner räumlicher Ketten mit rotativen und zylindrischen Gelenken

16:00 – 16:15 Uhr: Kaffeepause

16:15 – 18:15 Uhr: Session IV: BioMed Anw. (Leitung: Francisco Geu Flores)

Sebastian Kurz (Hochschule Koblenz) und Harold Schreiber

Kinematische Auslegung von Handprothesen - wie der 3D-Druck die Grenzen der Prothesentechnik neu definiert

Christina Insam (TU München)

Real-Time Substructuring (RTS) zum Testen von Fußprothesen

Nora-Sophie Staufenberg (TU München), Johanna Vielemeyer, Eric Griebbach, Roy Müller und Daniel Rixen

Vergleichende Betrachtungen von Bewegungen und Bewegungsmustern zwischen Menschen und humanoiden Robotern

Anna Sondermann, Francisco Geu Flores, Harald Hefter, Marcus Jäger, Andrés Kecskeméthy, Dominik Raab, Mario Siebler, Dörte Zietz und Silke Gegenbauer (ITBB GmbH, Univ. Duisburg-Essen)

Konzept einer interdisziplinären GUI für die Befundung und Diagnose von Gangstörungen bei der Therapieplanung von Schlaganfallpatienten — das RehaBoard-Projekt

Ab 20:00 Uhr: Gemeinsames Abendessen



5. IFToMM D-A-CH Konferenz 2019

26./27. Februar 2019, RWTH Aachen

Programm 27. Februar

8:00 – 10:00 Uhr: Session V: Mikrosysteme & Nachgiebige Systeme
(Leitung: René Theska)

Paul Motzki (ZeMa, Saarbrücken), Steffen Hau und Stefan Seelecke

Smart and Energy-efficient Actuators based on Shape Memory Alloys and Dielectric Elastomers

Erich Wehrle (Freie Univ. Bozen-Bolzano), Johannes Achleitner und Joseph Reinisch
Optimale Synthese nachgiebiger Mechanismen für die Robotik

Luzius Kronig (EPF Lausanne), Philipp Hoerler, Jean-Paul Kneib, Mohamed Bouri und Hannes Bleuler

Herausforderungen in Design und Kalibration für hoch-präzise Mikroroboter für Spektroskopie Messungen von Galaxien

Florian Weigert (TU Ilmenau) und René Theska

Entwicklung einer Wechselschnittstelle für Tools in Nanofabrikationsmaschinen mit hochreproduzierbarer Lagedefinition

10:00 – 10:15 Uhr: Kaffeepause

10:15 – 12:15 Uhr: Session VI: Dynamik & Regelung
(Leitung: Christoph Woernle)

Yannik Peters (Porsche AG)

Ein seitenkraftbasierter, zentraler Querdynamikregler für Sportwagen

Niklas Stannartz (TU Dortmund), Avraam Tolmidis, Oliver Maria Kind und Torsten Bertram

Vergleich von Graphen- und Kalman-Filter-basierten Verfahren zur verbesserten Eigenbewegungsschätzung auf Basis detektierter statischer Objekte

Michael Stoltmann (Univ. Rostock) und Christoph Woernle

Feedforward Control of a Crane Manipulator with Four Load Chains

Andreas Schulze (Univ. Rostock), Christoph Woernle und János Zierath

Weiterentwicklung eines Model-Update-Ansatzes flexibler Mehrkörpersysteme



5. IFToMM D-A-CH Konferenz 2019

26./27. Februar 2019, RWTH Aachen

12:15 – 13:15 Uhr: Mittagessen

13:30 – 15:00 Uhr: Session VII: Dynamik & Regelung (Leitung: Torsten Bertram)

Christian Gollee (TU Dresden)

Parameteridentifikation eines dynamischen mechanischen Systems mittels Frequenzganganalyse und rekursiver Partikelschwarmoptimierung für die Trajektorienplanung

Janosch Luttmer (Univ. Duisburg-Essen), Sebastian Reicherts und Dieter Schramm

Analyse der Benutzer- und Softwareanforderungen bei der Entwicklung eines modularen Software-Frameworks für mechanische Systeme am Beispiel von Fahrdynamikmodellen

Mirko Waldner (TU Dortmund), Maximilian Krämer und Torsten Bertram

Ein Hardware-in-the-Loop-Prüfstand zur Verifikation und Validierung von Matrixscheinwerfern

15:00 – 15:15 Uhr: Verabschiedung durch Tobias Bruckmann

16:00 – 17:30 Uhr: Optional: Domführung

Beiträge

Claudio Villegas, Mathias Hüsing, Burkhard Corves Seite 10 <i>Synthesis of Mechanisms Based on their Joint-Forces</i> https://doi.org/10.17185/dupublico/48189
Alexandra Mercader, Anian Leyerer, Franz Irlinger, Tim C. Lueth Seite 12 <i>Deep Learning zur Viergelenksynthese unter Verwendung der Fourier-Koeffizienten der Koppelkurve</i> https://doi.org/10.17185/dupublico/48190
Thorsten Speicher, Maik Berger Seite 17 <i>Methoden zur Auswahl und Synthese von Bewegungsgesetzen für mechatronische Systeme zur effizienten Prozesssteigerung</i> https://doi.org/10.17185/dupublico/48191
Liang Chen, Philipp Wabnitz, Maik Berger, Fan Yin Seite 24 <i>Design und Konstruktion eines neuen Konzepts für Mehrteilgreifer</i> https://doi.org/10.17185/dupublico/48192
Michael Gerbl, Michael Pieber, Johannes Gerstmayr Seite 31 <i>Reconfiguration of Modular Robots with triangular structure</i> https://doi.org/10.17185/dupublico/48193
Veronika Zumpe, Johannes Horeis, Mathias Hüsing, Burkhard Corves Seite 39 <i>Pre-Processing für taskbasierte Kinematik-Synthese</i> https://doi.org/10.17185/dupublico/48194
Justus Kopp, Bertold Bader, Frank Woittennek Seite 47 <i>Online-Optimierung von Geschwindigkeitsprofilen entlang vorgegebener Bahnen unter Berücksichtigung von Achsruckbeschränkungen</i> https://doi.org/10.17185/dupublico/48195
Martin Sereinig, Johannes Gerstmayr, Martin Pfurner Seite 55 <i>Kollisionsvermeidung und Pfadplanung im diskreten Gelenkwinkelraum redundanter Manipulatoren</i> https://doi.org/10.17185/dupublico/48196
Amirreza Shahidi, Tim Detert, Mathias Hüsing, Burkhard Corves Seite 57 <i>Agile Montagesystem durch Konsens von Roboter- und Sensornetzwerk – Eine Einführung zur Industrie 4.0</i> https://doi.org/10.17185/dupublico/48197
Philipp Wabnitz, Andreas Heine, Maik Berger Seite 59 <i>Vergleich von haptischen Steuerungsvarianten zur manuellen Bedienung von Robotersystemen</i> https://doi.org/10.17185/dupublico/48198
Tobias Bruckmann, Arnim Spengler Seite 66 <i>Simulation automatisierter Bauprozesse unter Einsatz von Seilrobotern</i> https://doi.org/10.17185/dupublico/48199

Bertold Bongardt	Seite 68
<i>Eine analytische Lösung der inversen Kinematik allgemeiner räumlicher Ketten mit rotativen und zylindrischen Gelenken</i>	
https://doi.org/10.17185/dupublico/48200	
Sebastian Kurz, Harold Schreiber	Seite 69
<i>Kinematische Auslegung von Handprothesen – wie der 3D-Druck die Grenzen der Prothesentechnik neu definiert</i>	
https://doi.org/10.17185/dupublico/48201	
Christina Insam, Daniel Rixen	Seite 70
<i>Real-Time Substructuring (RTS) zum Testen von Fußprothesen</i>	
https://doi.org/10.17185/dupublico/48204	
Nora-Sophie Staufenberg, Johanna Vielemeyer, Eric Griebach, Roy Müller, Daniel Rixen	Seite 72
<i>Vergleichende Betrachtungen von Bewegungen und Bewegungsmustern zwischen Menschen und humanoiden Robotern</i>	
https://doi.org/10.17185/dupublico/48205	
Anna Sondermann, Francisco Geu Flores, Harald Hefter, Marcus Jäger, Andrés Kecskeméthy, Dominik Raab, Mario Siebler, Dörte Zietz, Silke Gegenbauer	Seite 73
<i>Konzept einer interdisziplinären GUI für die Befundung und Diagnose von Gangstörungen bei der Therapieplanung von Schlaganfallpatienten - das RehaBoard-Projekt</i>	
https://doi.org/10.17185/dupublico/48207	
Paul Motzki, Steffen Hau, Stefan Seelecke	Seite 75
<i>Smart and Energy-efficient Actuators based on Shape Memory Alloys and Dielectric Elastomers</i>	
https://doi.org/10.17185/dupublico/48209	
Erich J. Wehrle, Johannes Achleitner, Joseph Reinisch	Seite 77
<i>Optimale Synthese nachgiebiger Mechanismen für die Robotik</i>	
https://doi.org/10.17185/dupublico/48210	
Luzius Kronig, Philipp Hörler, Hannes Bleuler, Jean-Paul Kneib, Mohamed Bouri	Seite 79
<i>Herausforderungen in Design und Kalibration für hoch-präzise Mikroroboter für Spektroskopie Messungen von Galaxien</i>	
https://doi.org/10.17185/dupublico/48212	
Florian Weigert, René Theska	Seite 90
<i>Entwicklung einer Wechselschnittstelle für Tools in Nanofabrikationsmaschinen mit hochreproduzierbarer Lagedefinition</i>	
https://doi.org/10.17185/dupublico/48214	
Yannik Peters	Seite 94
<i>Querdynamikregler für Sportwagen</i>	
https://doi.org/10.17185/dupublico/48217	
Niklas Stannartz, Avraam Tolmidis, Oliver Maria Kind, Torsten Bertram	Seite 103
<i>Vergleich von Graphen- und Kalman-Filter-basierten Verfahren zur verbesserten Eigenbewegungsschätzung auf Basis detektierter statischer Objekte</i>	
https://doi.org/10.17185/dupublico/48218	

Michael Stoltmann, Christoph Woernle, Pascal Froitzheim, Normen Fuchs, Wilko Flügge	Seite 105
<i>Feedforward Control of a Crane Manipulator with Four Load Chains</i>	
https://doi.org/10.17185/dupublico/48219	
Andreas Schulze, Christoph Woernle, János Zierath	Seite 107
<i>Weiterentwicklung eines Model-Update-Ansatzes flexibler Mehrkörpersysteme</i>	
https://doi.org/10.17185/dupublico/48220	
Christian Gollee	Seite 109
<i>Parameteridentifikation eines dynamischen mechanischen Systems mittels Frequenzganganalyse und rekursiver Partikelschwarmoptimierung für die Trajektorienplanung</i>	
https://doi.org/10.17185/dupublico/48221	
Janosch Luttmer, Sebastian Reicherts, Dieter Schramm	Seite 110
<i>Analyse der Benutzer- und Softwareanforderungen bei der Entwicklung eines modularen Software-Frameworks für mechanische Systeme am Beispiel von Fahrdynamikmodellen</i>	
https://doi.org/10.17185/dupublico/48222	
Mirko Waldner, Maximilian Krämer, Torsten Bertram	Seite 118
<i>Ein Hardware-in-the-Loop-Prüfstand zur Verifikation und Validierung von Matrixscheinwerfern</i>	
https://doi.org/10.17185/dupublico/48223	

Synthesis of Mechanisms Based on their Joint-Forces

Claudio Villegas, RWTH Aachen University, Institute of Mechanism Theory, Machine Dynamics and Robotics, 52074 Aachen, Germany and University of Bio-Bio, Department of Mechanical Engineering, Concepcion, Chile. villegas@igmr.rwth-aachen.de

Mathias Hüsing, RWTH Aachen University, Institute of Mechanism Theory, Machine Dynamics and Robotics, 52074 Aachen, Germany.

Burkhard Corves, RWTH Aachen University, Institute of Mechanism Theory, Machine Dynamics and Robotics, 52074 Aachen, Germany.

Abstract

The design of mechanisms is typically conducted by sequentially following steps intended to [1]:

1. Determine the topology of the kinematic chain underlying the mechanical structure.
2. Define the link geometry based on the task/workspace requirements.
3. Determine the structural dimensioning of links and joints to meet the static loads requirements.
4. Determine the structural dimensioning to include the inertia effects of links, joints and the manipulated object.
5. Determine the elastodynamic dimensioning of the entire structure to ensure its eigenfrequencies avoid the operational speed range.
6. Select the actuators and their mechanical transmissions for the given task.

However, if the dynamic or elastodynamic behavior is unacceptable, then redimensioning is required, which means returning to step three. Moreover, when motors are selected (step 6), a new elastodynamic analysis is necessary. Thus, this process is iterative and sometimes several loops are needed to accomplish an appropriate design. For high-speed machinery, this is specially challenging. High speed increases the nonlinearities, dynamic reaction forces, vibration and wearing [2]. Therefore, considering the dynamic behavior in the first stage of the design would increase the efficiency of the process, by obtaining an earlier dynamic performance estimation.

Several authors have addressed the problem of designing mechanisms with good dynamic behavior from the very beginning of the design process, either by optimization [3, 4] or by layered approaches [5]. Those works use as constraint the dynamic characteristics as the drive torque, the energy consumption, the velocity and/or acceleration of the limbs, but very little attention is paid to reaction forces.

The aforementioned motives induced the authors to start developing a mechanism synthesis method that minimizes the dynamic reaction forces in the joints of the mechanisms (Joint-Forces). The proposed idea defines a first stage, where the synthesis of Function Generator Four-Bar linkages based on the Dead-Center synthesis method [6], with Joint-Forces minimization, was studied parametrically. Based on the auxiliary angle presented in [6], the equations were parametrized for the case when the rocker moves between two positions, the crank rotates with constant angular velocity and the mass and inertia of the links is considered proportional to the link lengths and the squared link lengths, respectively. As objective function was used the peak Joint-Force. It was found that the Maximum Joint-Force function is convex, and hence minimizable. However, one of the difficulties associated to this study is that, in some cases, the resulting optimal design point does not fulfill the classical optimality conditions, lying on a discontinuity of the function derivative (Figure 1). This inconvenient was overcome by using derivative non-dependent resolution methods.

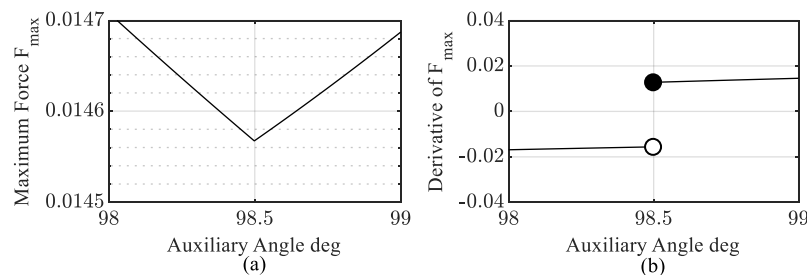


Figure 1. (a) Maximum Joint-Forces as function of the Auxiliary Angle. (b) Derivative of the Maximum Joint-Force with respect to the auxiliary angle.

As an example of the methodology, the synthesis of a Four-Bar linkage that moves a high inertia amount in the rocker between two positions will be exposed. It will also be presented the challenges to extend the method to any four-bar linkage with more than two precision points, and then to other mechanisms with more than four bars or even with more degrees of freedom.

References

- [1] Angeles, J.; Park, F. C.: *Performance Evaluation and Design Criteria*. In: Siciliano, B.; Khatib, O. (Hrsg.): Springer handbook of robotics. Springer Berlin (u.a.) 2008. ISBN 978-3-540-23957-4. S. 229–244.
- [2] Rayner, R.; Sahinkaya, M. N.; Hicks, B.: *Combining Inverse Dynamics With Traditional Mechanism Synthesis to Improve the Performance of High Speed Machinery*. In: Proceedings of the ASME dynamic systems and control conference 2008, ASME 2008 Dynamic Systems and Control Conference. 20./22. Oktober 2008, Ann Arbor, Michigan, USA. American Society of Mechanical Engineers New York, N.Y. 2009. ISBN 978-0-7918-4335-2. S. 599–606.
- [3] Schwarzfischer, F.; Hüsing, M.; Corves, B.: *The Dynamic Synthesis of an Energy-Efficient Slider-Crank-Mechanism*. In: Proceedings of the Second International Symposium of Mechanism and Machine Science, Proceedings of the Second International Symposium of Mechanism and Machine Science (ISMMS-2017). 11-14 September, Baku, Azerbaijan. Baku 2017. S. 156–163.
- [4] Conte, F. L.; George, G. R.; Mayne, R. W.; Sadler, J. P.: *Optimum Mechanism Design Combining Kinematic and Dynamic-Force Considerations*, Journal of Engineering for Industry 97 (1975), Nr. 2. S. 662, doi: 10.1115/1.3438631.
- [5] Rayner, R. M.C.; Sahinkaya, M. N.; Hicks, B.: *Improving the design of high speed mechanisms through multi-level kinematic synthesis, dynamic optimization and velocity profiling*, Mechanism and Machine Theory 118 (2017). S. 100–114, doi: 10.1016/j.mechmachtheory.2017.07.022.
- [6] Verein Deutscher Ingenieure: *Crank and Rocker Mechanisms*: VDI 2130, Duesseldorf: VDI-Verlag GmbH, 1984

Deep Learning zur Viergelenksynthese unter Verwendung der Fourier-Koeffizienten der Koppelkurve

Deep Learning for Planar Four-bar Linkage Synthesis

Alexandra Mercader, Anian Leyerer, Franz Irlinger, Tim C. Lueth, Technische Universität München, Lehrstuhl für Mikrotechnik und Medizingerätetechnik, 85748 Garching, Deutschland, alexandra.mercader@tum.de

Kurzfassung

Dieser Artikel beschreibt einen neuen Ansatz für die Synthese von Viergelenkketten. Die Fourier-Koeffizientenanalyse zur Beschreibung geschlossener Kurven erlaubt das Verständnis einiger Besonderheiten der Koppelkurven. Diese Analyse ist deshalb interessant, weil sie es ermöglicht, die Form jeder Kurve zu speichern. Diese Form kann dann als Merkmal für den Aufbau eines neuronalen Netzwerks verwendet werden. Dieses Netzwerk wird dann trainiert, um die Parameter des zugrundeliegenden Mechanismus zu ermitteln. Diese Regression zwischen den Fourier-Koeffizienten der geschlossenen Kurve und den Abmessungen des Mechanismus wird ausführlich beschrieben und ausgewertet, um den Mehrwert dieses Ansatzes für die kinematische Synthese der Viergelenkkette zu verstehen.

Abstract

This article describes a new approach for the synthesis of four-bar mechanisms. Fourier coefficient analysis for the description of closed curves provides an understanding of some of the particularities of coupler curves. This analysis is relevant as it allows the shape of each curve to be stored in memory. This form can then be used as a characteristic for neural network modeling. This network is trained to retrieve the parameters of the original mechanism from the Fourier coefficients of the curve. This regression between the coefficients of the closed curve and the dimensions of the mechanism is discussed in detail and evaluated to understand the benefits of this approach for the kinematic synthesis of the planar four-bar linkage.

1 Einleitung

Die kinematische Synthese von Viergelenk-Mechanismen wird auf verschiedene Weise gelöst. In diesem Artikel wird nur die Maßsynthese einer Kurbelschwinge betrachtet. Die Methoden zur Synthese von Mechanismen aus Lagen oder aus Punktlagen variieren, können aber miteinander verglichen werden, da zwei Punkte zu einer Lage assoziiert werden können und eine Lage im Falle eines ersten Ansatzes auf einen Punkt reduziert werden kann [1]. Die am häufigsten verwendete Methode ist die grafische Synthese [2, 3]. In der ebenen Synthese ist bekannt, dass maximal fünf Lagen berücksichtigt werden können. Für eine größere Anzahl an Lagen werden Optimierungsmethoden verwendet, um eine Lösung für den gewünschten Pfad zu approximieren [4, 5]. Eine weitere Möglichkeit besteht darin, Atlanten zu beobachten und mit bloßem Auge oder per Computer den am besten geeigneten Mechanismus zu wählen [6]. In aktuellen Forschungsarbeiten wurden bereits Regressionstests zwischen den Fourier-Koeffizienten der Koppelkurve und den Parametern der Viergelenkkette durchgeführt [7, 8].

Die Versuche, die durchgeführt wurden, um eine Regression zwischen den Fourier-Koeffizienten und den Parametern des Mechanismus zu ermitteln, liefern jedoch keine Grundlage für einen Leistungsvergleich von Deep Learning Algorithmen für die mechanische Synthese. Darüber hinaus wurde die Bedeutung von Fourier-

Koeffizienten bei Koppelkurven ebenfalls nicht untersucht.

Zunächst werden Fourier-Koeffizienten analysiert und eine Klassifizierung durchgeführt, um die verschiedenen Kurventypen aus der Perspektive der Fourier-Koeffizienten besser zu verstehen. Dann wird der Grund für die Auswahl des Algorithmus näher erläutert und die Auswirkung der Variation von Fourier-Koeffizienten auf die Maße des Mechanismus veranschaulicht. In einem zweiten Teil werden die allgemeinen Ergebnisse vorgestellt und in einem letzten Teil wird die Verwendung des Algorithmus an einem Beispiel dargestellt. Die in diesem Artikel verwendete Nomenklatur und Bedingungen sind in Bild 1 definiert.

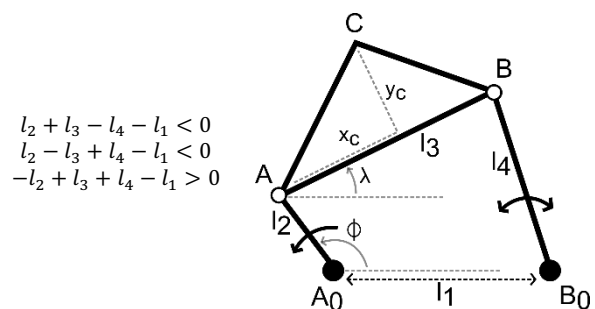


Bild 1 Nomenklatur und Bedingungen für die Maße der Viergelenkkette

2 Fourier-Koeffizienten für geschlossene Koppelkurven

2.1 Zerlegung in Fourier-Koeffizienten

Die Koppelkurve wird parametrisch definiert, wie in [7]. Zu diesem Zweck werden 100 äquidistante Werte des ϕ -Winkels zwischen 0 und 2π gewählt, um die diskreten Werte der Koppelkurve C zu berechnen. Die parametrische Gleichung wird durch die Gleichungen zum Schließen des Mechanismus abgeleitet.

$$C = \frac{l_2}{l_1} e^{j\phi} + \frac{1}{l_1} (x_c + jy_c) e^{j\lambda} \quad (1)$$

Die Fourier-Koeffizienten einer geschlossenen Kurve ermöglichen es, die reinen Formen der Kurven zu vergleichen, ohne ihre Ausrichtungen, Positionen und Größenordnungen zu berücksichtigen. Für Mechanismen reduzieren Fourier-Koeffizienten auch die räumlichen Dimensionen der Kurvenparameter, indem nur die tiefen Frequenzen beibehalten werden. In [9] sind die niedrigen Frequenzen ausreichend, um ein Beispiel für eine Koppelkurve zu rekonstruieren. Durch eine mathematische Technik namens Hauptkomponentenzerlegung [10] kann die tatsächliche Größe des Raums des Fourier-Koeffizienten einer Viergelenkkette jedoch genauer ermittelt werden.

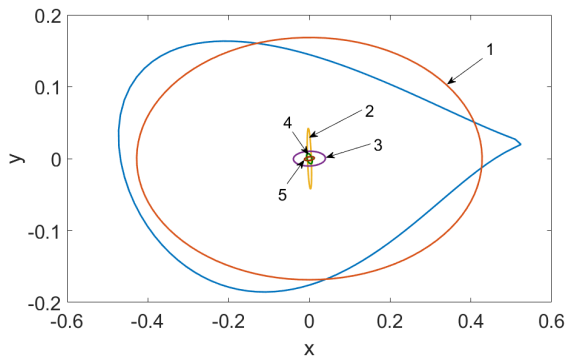


Bild 2 Erste fünf Harmonische einer Koppelkurve einer Viergelenkkette und die ursprüngliche Kurve (blau)

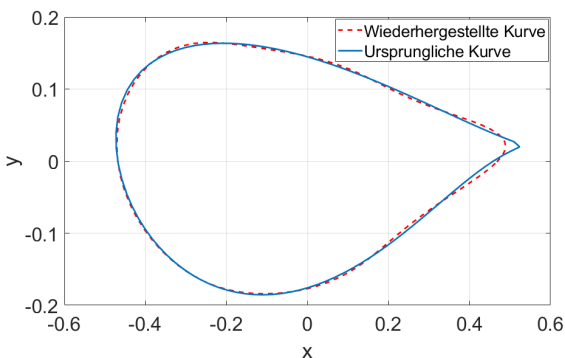


Bild 3 Wiederhergestellte Kurve aus dem fünf Harmonischen von der ursprüngliche Koppelkurve

Vor der Analyse der Koeffizienten wird die Koppelkurve normiert und parametrisiert, so dass nur die Geometrie verglichen wird. Dabei beeinflussen Orientierung, Position, Skalierung und Parametrisierung der Kurve die Fourier-Koeffizienten. Da jede Kurve eine unendliche Anzahl an möglichen Parametrisierungen hat, ist es wichtig, diesen Schritt durchzuführen und für jede Kurve eine Parametrisierung zu definieren. Die gewählte Parametrisierung ist so ausgelegt, dass die Geschwindigkeit des Bahnverlaufes konstant ist, mit einer Periode von 1. Die Position, Ausrichtung und Skalierung werden ebenfalls ignoriert, da das Gestell neu ausgerichtet und verschoben werden kann und seine Maße proportional zur Größe des Kurvenverlaufs sind. Ein Beispiel für eine normierte Kurve und ihre jeweiligen Koeffizienten ist in Bild 2 dargestellt.

Jede Harmonische ist eine Ellipse, deren Hauptorientierung und Länge der Achsen durch Fourier-Koeffizienten leicht überprüfbar sind. Wie in der Abbildung zu sehen ist, schrumpfen die Ellipsen zunehmend. Tatsächlich sind nur die ersten fünf Harmonischen wichtig, um die Kurve darzustellen. Die aus den fünf Harmonischen wiederhergestellte Kurve ist in Bild 3 dargestellt.

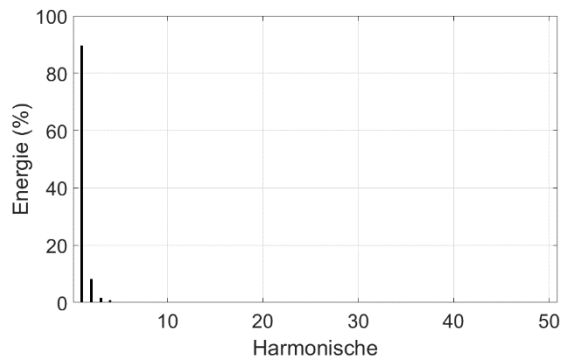


Bild 4 Abbildung der 50 ersten Harmonischen des Datensatzes der Koppelkurven

2.2 Hauptkomponentenanalyse

Um zu rechtfertigen, dass nur fünf Harmonische benötigt werden, erlaubt uns die Hauptkomponentenanalyse, die räumliche Dimension der Fourier-Koeffizienten der Koppelkurven zu untersuchen. Die Maße des Datensatzes variieren zwischen $1/3$ und 3 , und l_1 ist immer gleich 1 . Der Beitrag jedes der ersten 50 Eigenwerte der Kovarianzmatrix des Koeffizientensatzes ist in Abbildung 4 dargestellt. Dabei ist ersichtlich, dass fünf Dimensionen ausreichend sind, um diesen Satz darzustellen. Dadurch werden die Koeffizienten auf 10 Werte reduziert, die fünf positiven und negativen Frequenzen entsprechen, wobei negative Frequenzen ebenfalls eine Rolle bei der Darstellung der Kurve spielen, aber keine physikalische Bedeutung haben. Diese Analyse zeigt auch, dass die Koppelkurven nur Frequenzen zwischen dem Ein- und Fünffachen der Getriebefrequenz aufweisen.

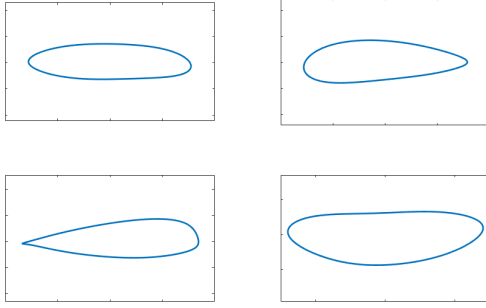


Bild 5 Klassenzentren nach dem k-Means-Algorithmus (Rückwärts-Fourier-Transformation)

2.3 Klassifizierung der Koeffizienten

Eine abschließende Analyse besteht darin, die Koeffizienten unüberwacht zu klassifizieren. Dazu wird der k-Means-Algorithmus [11] verwendet, um die verschiedenen Kurventypen zu erfassen, die im Datensatz vorhanden sind. Die Ergebnisse werden für 2, 3, 4 und 5 Klassen berücksichtigt. Dabei zeigt sich, dass eine Anzahl von 4 Klassen ideal ist, da jeder Klasse eine einzigartige Form zugewiesen werden kann. Bild 5 zeigt die Klassenzentren nach dem k-Means-Algorithmus, die zufällig mit einer Probe des Datensatzes initialisiert wurden. Die Ergebnisse zeigen, dass die Koppelkurven vier verschiedene Formen haben: mit der Spitze nach rechts oder nach links und mit einer ziemlich flachen Trajektorie oben oder unten. Natürlich kann es komplexere Trajektorien mit Doppel- oder Tripelpunkten geben. Diese Geometrien sind in diesem Datensatz jedoch nicht dominant. Das bedeutet, dass der Algorithmus nicht in der Lage sein wird, solche Kurven zu approximieren.

2.4 Nachteile der Datenbank-Suche

Wie bereits in [9] erläutert, ermöglicht die erste Anwendung der Fourier-Koeffizienten Theorie in der Industrie eine schnelle Suche in einer Datenbank nach der nächstgelegenen Kurve, die den gewünschten Punktlagen entspricht. Dann werden die Maße des Mechanismus optimiert, um eine genauere Annäherung an die gewünschte Kurve zu erhalten. Eine einfache Änderung der Abmessungen einer Viergelenkkette führt jedoch oft zu einer vollständigen Verformung der Kurve. Dadurch wird die Optimierung komplexer, da das Verhältnis zwischen der Kurve und den Abmessungen des Mechanismus nicht linear ist. Dieses Verhältnis wurde von Beyer [3] bereits in der Formel der impliziten Gleichung der Koppelkurve nachgewiesen. Um die Nichtlinearität dieser Beziehung zu visualisieren, veranschaulicht Bild 6 die Variation des euklidischen Abstands zwischen einem Fourier-Koeffizientenvektor und den 50 nächsten, die mit den Maßen der Viergelenkkette linear erzeugt wurden. Die

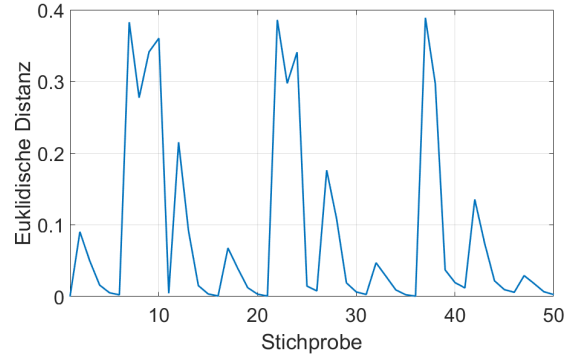


Bild 6 Euklidische Distanz zwischen einer Einzelprobe und den nächsten fünfzig Proben

euklidische Distanz zwischen zwei Vektoren F_1 und F_2 mit n Koeffizienten wird auf diese Weise einfach berechnet:

$$D(F_1, F_2)^2 = \sum_{i=1}^n |F_{1i} - F_{2i}|^2 \quad (2)$$

Da die Beziehung zwischen Maße und Fourier-Koeffizienten nicht linear ist und die Anzahl der verfügbaren Daten unbegrenzt ist, eignen sich neuronale Netze besonders, um die Abhängigkeit zu approximieren und das Problem der Maßsynthese zu lösen.

3 Algorithmus und Ergebnisse

3.1 Beschreibung des Verfahrens

Bild 7 illustriert die verschiedenen Schritte des verwendeten Verfahrens. Das neuronale Netz erhält als Eingabe die Fourier-Koeffizienten der normierten, zentrierten und reduzierten Kurve. Die Maße des Mechanismus werden am

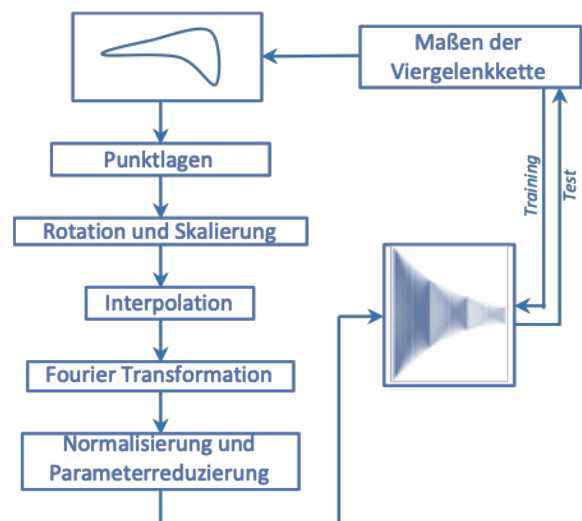


Bild 7 Blockdiagramm des verwendeten Verfahrens

Output für das Training verwendet und während der Testphase mit den gewünschten Dimensionen verglichen. Zusätzlich wird die den Ausgangsmaßen entsprechende Kurve analysiert, falls diese existiert. Um die Performance dieses Algorithmus zu bestimmen, fasst die Ergebnistabelle 1 die Berechnungen der durchschnittlichen Distanz zwischen der gewünschten und der erhaltenen Kurve einer Testprobe zusammen. Der berechnete Abstand E_c ist die kleinere Summe der Abstände zwischen den einzelnen Punkten auf den beiden Kurven. Dadurch bleibt die Reihenfolge der Punkte in der Berechnung erhalten, die wie folgt berechnet wird:

$$E_c = \frac{1}{l} \min_j \left(\sum_{i=1}^l |C_t(i) - C_o(j+i)| \right) \quad (3)$$

C_t ist dabei die gewünschte Kurve, C_o die ausgegebene Kurve und l der Anzahl an Punktlagen.

Das neuronale Netz wird jedoch nur durch eine reine Backpropagation optimiert, die den Fehler zwischen dem gewünschten Output t und dem durch Forward Propagation erzeugten Output o berücksichtigt. Der Fehler ist in diesem Fall:

$$P = \frac{1}{N} \sum_{i=1}^N (t_i - o_i)^2 \quad (4)$$

Da die Maße des Mechanismus immer positiv sind und die verwendete Aktivierungsfunktion die Sigmoidfunktion ist, ist es wichtig, die beiden Fälle zu trennen. Darüber hinaus müssen aufgrund der beiden möglichen Konfigurationen einer Kurbelschwinge auch zwei weitere berücksichtigt werden. Insgesamt sind vier Konfigurationen des Mechanismus möglich. Bild 8 zeigt die verschiedenen Konfigurationen.

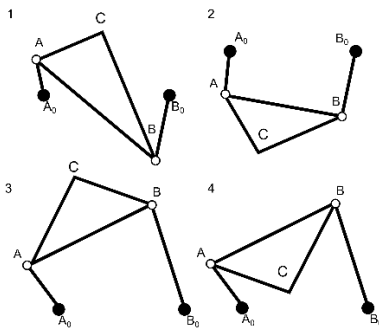


Bild 8 Vier mögliche Konfigurationen für vier verschiedene Netze

3.2 Ergebnisse

Für jede Konfiguration sind die Ergebnisse von 1, 2 und 3 verborgenen Schichten in Bild 9 dargestellt. Die Ergebnisse variieren je nach Anzahl der verborgenen Schichten. Der minimale quadratische Fehler liegt bei etwa 0,1 für die Maße des Mechanismus, was auch eine andere Kurve als die gewünschte erzeugen kann. Das Interessanteste ist, den Fehler auf der Kurve zu betrachten, wie im vorherigen Absatz erläutert. Für drei verborgene

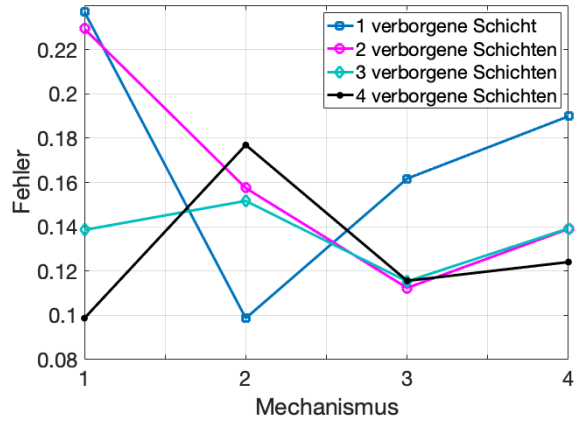


Bild 9 Fehler P für die verschiedenen Mechanismen 1 bis 4

Schichten ist der Fehler P für jede Konfiguration des Mechanismus am stabilsten. Tabelle 1 zeigt den Fehler für drei verborgenen Schichten.

Tabelle 1 Fehler E_c für drei verborgenen Schichten

Mechanismus		1	2	3	4
Fehler	Position (skaliert)	0.1013	0.1164	0.3536	0.1254

Der Fehler wird auf der normierten Kurve berechnet, bei der die Breite auf 1 reduziert wird. Dieser Fehler setzt auch voraus, dass die Reihenfolge der Punkte eingehalten wird, was die Größenordnung der erhaltenen Werte erklärt. Für zwei verborgene Schichten kann der Algorithmus jedoch eine Kurve mit einer Genauigkeit von etwa einem Zehntel der Gesamtbreite der Kurve approximieren.

4 Zusammenfassung

Zusammenfassend ist festzustellen, dass die Nichtlinearität der Beziehung zwischen den Abmessungen einer Kurbelschwinge und ihrer Koppelkurve mit Hilfe von Deep Learning Verfahren angenähert werden kann. Diese Forschung wird nur an einem Viereck-Mechanismus durchgeführt, um zunächst herauszufinden, wie präzise dieser Ansatz ist und wie brauchbar dieses Verfahren ausfällt. Obwohl die Ergebnisse nicht mit anderen Methoden verglichen wurden, haben nur wenige Arbeiten eine detaillierte Studie über die Verwendung dieses Algorithmus durchgeführt. Dadurch ergeben sich neue Anwendungen, wie zum Beispiel die Synthese von Mechanismen aus Punktlagen, die mit optischen Instrumenten gemessen werden.

Durch diese Ergebnisse ergeben sich jedoch viele Forschungsmöglichkeiten. Zwar sind neuronale Netze sehr beliebt, aber andere Deep Learning Algorithmen, die die Mechanismus-Synthese besser repräsentieren können, könnten getestet werden, um die Synthese von Mechanismen im Allgemeinen zu beschleunigen und zu automatisieren.

5 Literatur

- [1] Alt, H.: *Über die Erzeugung gegebener ebener Kurven mit Hilfe des Gelenkviereckes.*, *Z. angew. Math. Mech.* (1923) 3-1, S. 13–19.
- [2] McCarthy, J. M.; Soh, G. S., *Geometric Design of Linkages*, 2nd ed. New York, NY: Springer Science+Business Media LLC, 2011.
- [3] Beyer, R., *Kinematische Getriebesynthese*. Berlin, Heidelberg: Springer Berlin Heidelberg, 1953.
- [4] Cabrera, J. A.; Simon, A.; Prado, M.: *Optimal synthesis of mechanisms with genetic algorithms.*, *Mechanism and Machine Theory*. (2002) 37-10, S. 1165–1177.
- [5] Lin, W.-Y.: *A GA–DE hybrid evolutionary algorithm for path synthesis of four-bar linkage.*, *Mechanism and Machine Theory*. (2010) 45-8, S. 1096–1107.
- [6] Hrones, J. A.; Nelson, G. L., *Analysis of the Four-Bar Linkage: Its Application to the Synthesis of Mechanisms*: The MIT Press, 1951.
- [7] Li, X.; Chen, P.: *A Parametrization-Invariant Fourier Approach to Planar Linkage Synthesis for Path Generation.*, *Mathematical Problems in Engineering*. (2017).
- [8] Vasiliu, A.; Yannou, B.: *Dimensional synthesis of planar mechanisms using neural networks: application to path generator linkages.*, *Mechanism and Machine Theory*. (2001) 36-2, S. 299–310.
- [9] McGarva, J.; Mullineux, G.: *Harmonic representation of closed curves.*, *Applied Mathematical Modelling*. (1993) 17-4, S. 213–218.
- [10] Pearson, K.: *On lines and planes of closest fit to systems of points in space.*, *Philosophical Magazine*. (1901) 2, S. 559–572.
- [11] Forgy, E.: *Cluster analysis of multivariate data: efficiency versus interpretability of classifications.*, *Biometrics*. (1965) -21, S. 768–780.

Methoden zur Auswahl und Synthese von Bewegungsgesetzen für mechatronische Systeme zur effizienten Prozesssteigerung

Thorsten Speicher, M. Eng., *Technische Universität Chemnitz, Professur Montage- und Handhabungstechnik, 09126 Chemnitz, Deutschland*, thorsten.speicher@mb.tu-chemnitz.de

Univ.-Prof. Dr.-Ing. Maik Berger, *Technische Universität Chemnitz, Professur Montage- und Handhabungstechnik, 09126 Chemnitz, Deutschland*, maik.berger@mb.tu-chemnitz.de

Kurzfassung

Der zunehmende Trend zur Flexibilisierung in engem Zusammenhang mit Industrie 4.0 gewinnt im industriellen Umfeld immer stärker an Bedeutung. Um dem Konkurrenzdruck in diesem Bereich standhalten zu können, sind Unternehmen gezwungen, sich durch innovative Ansätze vom Wettbewerb abzuheben. In diesem Beitrag wird eine Vorgehensweise zur effektiven Optimierung von Bewegungsprofilen vorgestellt. Dazu wird die Auswahl bzw. Synthese von Bewegungsfunktionen unterstützt, die das komplette Leistungsvermögen der Maschine ausnutzen, um Taktzeiten zu reduzieren und folglich die Ausbringungsmenge zu erhöhen. Zudem bietet der Ansatz die Möglichkeit zur softwareseitigen Implementierung in eine SPS, um online produkt- oder prozessspezifische Bewegungsprofile zu optimieren.

Abstract

The increasing trend toward flexibility and industry 4.0 becomes more important in industrial environments. Because of the competitive pressure, the companies are forced to develop new and innovative approach setting themselves apart from the competition.

This paper presents a method for the effective optimization of motion profiles. The selection and synthesis of motion law are supported, that can use the entire performance of the processing machine to reduce the cycle time and to increase the output quantity. Moreover, this approach enables implementing in the PLC to realize an online optimization of product- or process-specific motion profiles.

1 Einleitung

Unter Betrachtung der betriebswirtschaftlichen Zwänge produzierender Unternehmenszweige in Bezug auf die permanente Erhöhung der Produktionsgeschwindigkeiten sowie dem gleichzeitigen Wunsch nach Flexibilisierung, bietet das Bewegungsdesign durch die Gestaltung optimaler Bewegungsfunktionen die Möglichkeit, Taktzeiten ohne den Einsatz von leistungsfähigeren Antriebskomponenten zu reduzieren. In der VDI Richtlinie 2143 aus dem Jahr 1980 [6] wird eine Arbeitsweise zur Gestaltung von Bewegungsverläufen durch die abschnittsweise Zusammensetzung von mathematischen Funktionen vorgestellt.

Aus einem Katalog können fest definierte, normierte Funktionen (Bewegungsgesetze) anhand vergleichbarer Gütekennwerte ausgewählt und mittels einer Transformation an die technologischen Randbedingungen angepasst werden. Aus aktuellen Veröffentlichungen zu den Themen „Optimierungen von Bewegungsgesetzen“ und „Entwicklung neuer Kennwerte“ geht hervor, dass diese Vorgehensweise vor allem für kurvengesteuerte Bewegungssysteme und Motion-Control-Systeme eine heute immer noch gängige Praxis ist [1], [2], [5]. Unter Einsatz leistungsfähiger Rechentechnik wurden spezielle Softwarelösungen entwickelt, die durch eine grafisch-interaktiv geprägte Arbeitsweise auch Möglichkeiten zur Bewegungsoptimierung bieten [3], [4]. Die angebotenen Softwareprodukte ermöglichen die Definition beliebig

vieler Stützpunkte mittels „PC-Maus“ in einem Diagramm-Editor, während zeitgleich, ähnlich einer Spline-Interpolation, die Berechnung eines geschlossenen Bewegungsverlaufs erfolgt. Eine Verschiebung der einzelnen Punkte (auch in den Ableitungen) unter gleichzeitiger Beobachtung der sich ergebenden Verläufe bezweckt eine Optimierung.

Bei detaillierter Analyse dieser heute gängigen Arbeitsweisen hinsichtlich ihrer Anwendbarkeit und Effizienz, vor allem in Bezug auf die flexible Optimierung von Bewegungsabläufen, ergeben sich neue Ansätze, die eine Erweiterung des Stands der Technik als sinnvoll erscheinen lassen.

Folgende Schwachstellen wurden identifiziert:

- Die fest definierten Funktionsverläufe der normierten Bewegungsgesetze mit ihren spezifischen Extrema der 1. und 2. Ableitung reizen in aller Regel die dominierenden prozess-, produkt- oder durch die Mechanik bedingten Bewegungsgrenzen (Geschwindigkeit und Beschleunigung) des mechatronischen Bewegungssystems nur einseitig aus. Einbußen in der Taktzeit und ferner in der Ausbringungsmenge sind die Konsequenz. Nur in bestimmten Ausnahmesituationen wird durch den Einsatz der angebotenen Bewegungsgesetze das Leistungsvermögen der Maschine annähernd vollständig ausgenutzt.

- Es fehlt eine effektive Methode zur effizienten Identifikation und Synthese taktzeitgünstiger Bewegungsgesetze für unterschiedliche Anforderungen.
- Das grafisch-interaktive Bewegungsdesign erlaubt oftmals eine Optimierung der Bewegungsprofile, sodass das Leistungspotential der Maschine weitgehend ausgenutzt werden kann. Diese aufwendige und zeitintensive Arbeitsweise setzt die Nutzung von kommerziellen Softwarelösungen voraus und ist aufgrund der händischen und simultanen Variation der einzelnen Punkte sowie der fehlenden systematischen Vorgehensweise nur für einen Experten zu empfehlen. Ein weiterer Nachteil ist die schlechte Reproduzierbarkeit der Ergebnisse.

Ziel des vorliegenden Beitrags ist es, eine Verfahrensweise zur Taktzeitoptimierung zu präsentieren, die die Auswahl und die Synthese von normierten Bewegungsgesetzen unterstützt, unter gleichzeitiger Berücksichtigung der zulässigen Bewegungsgrenzen des Systems. Im Vordergrund steht vor allem die einfache und strukturierte Anwendbarkeit, die es Anwendern ohne Expertenwissen und softwareunabhängig ermöglicht, gute Ergebnisse zu erzielen. Gleichzeitig soll die Voraussetzung zur softwareseitigen Implementierung in die Steuerung gegeben sein, um im Rahmen der Flexibilisierung auch individuell optimierte Bewegungsprofile im Online-Betrieb automatisiert zu berechnen und auszuführen.

In Bezug auf die Umsetzung der formulierten Ziele, werden neue Kennzahlen zur Charakterisierung der dominierenden Restriktionen des Bewegungssystems und der Bewegungsgesetze hinzugezogen. Diese Kennzahlen unterstützen neben der Auswahl von bekannten normierten Bewegungsgesetzen, auch die Synthese vorgeschlagener normierter Parameterfunktionen.

Die mit diesem Verfahren gewählten Funktionen reizen größtenteils die maximalen Bewegungsgrenzen des Systems aus und führen im Vergleich zu alternativen Optimierungsmethoden mit einem merklich reduzierten Zeitaufwand zu guten Optimierungsergebnissen.

Die Demonstration der Methoden erfolgt in diesem Beitrag anhand einer häufig im industriellen Umfeld eingesetzten Rast-in-Rast-Bewegung.

2 Grundlagen

In der Praxis kommen zur Realisierung solcher Bewegungsabläufe in der Regel kurvengesteuerte Mechanismen oder nach dem Master-Slave-Prinzip gesteuerte Motion-Control-Systeme zum Einsatz. Für eine konstante Antriebsgeschwindigkeit $\dot{x} = konst$ - bei Kurvengetrieben Antriebswinkelgeschwindigkeit und bei Motion-Control-Systemen generell die Geschwindigkeit der Leitachse - besteht der folgende Zusammenhang zwischen der Lage y , der Geschwindigkeit $\dot{y}$ und der Beschleunigung $\ddot{y}$ der Abtriebsbewegung für einen Abschnitt, mit der

Länge x_{ij} und dem Hub y_{ij} , und dem normierten Bewegungsgesetz $f(z)$ [6][7]:

$$y(x) = y_i + y_{ij} \cdot f(z) \quad (1)$$

$$\dot{y}(x) = \frac{y_{ij}}{x_{ij}} \cdot f'(z) \cdot \dot{x} \quad (2)$$

$$\ddot{y}(x) = \frac{y_{ij}}{x_{ij}^2} \cdot f''(z) \cdot \dot{x}^2 \quad (3)$$

$$\text{mit } z = \frac{x - x_i}{x_{ij}}$$

Durch Einsetzen der Beziehung

$$\dot{x} = \frac{x_{ij}}{t_{ij}} \quad (4)$$

in die Gleichungen (2) und (3), wobei t_{ij} die Taktzeit des Abschnitts darstellt und unter Berücksichtigung der normierten Kennwerte C_v und C_a ergeben sich die maximalen Bewegungsgrößen nach den Gleichungen:

$$\dot{y}_{\max} = \left| \frac{y_{ij}}{t_{ij}} \cdot C_v \right| \quad (5)$$

$$\ddot{y}_{\max} = \left| \frac{y_{ij}}{t_{ij}^2} \cdot C_a \right| \quad (6)$$

Die beiden Kennwerte C_v und C_a charakterisieren das betragsmäßige Maximum der jeweiligen Ableitung der normierten Bewegungsgesetze.

3 Kennwertentwicklung

Durch Einsetzen der max. zul. Geschwindigkeit $\dot{y}_{\max, zul}$ und Beschleunigung $\ddot{y}_{\max, zul}$ des Bewegungssystems in Gln. (5) und (6) ergeben sich die folgenden minimalen Taktzeiten für die Geschwindigkeitsbetrachtung

$$t_{ij, \min v} = \left| \frac{y_{ij}}{\dot{y}_{\max, zul}} \cdot C_v \right| \quad (7)$$

und für die Beschleunigungsbetrachtung

$$t_{ij, \min a} = \sqrt{\frac{y_{ij}}{\ddot{y}_{\max, zul}} \cdot C_a} \quad (8).$$

Ferner wird die min. zul. Taktzeit bestimmt mit:

$$t_{ij, \min} = \max(t_{ij, \min v}, t_{ij, \min a}) \quad (9)$$

Nun wird folgender Ansatz betrachtet:

$$t_{ij, \min v} = t_{ij, \min a} \quad (10)$$

Durch Einsetzen von Gln. (7) und (8) in (10) und bei gleichzeitiger Trennung der bewegungsgesetz- und der bewegungssystemabhängigen Größen resultiert:

$$\frac{C_a}{C_v^2} = \left| \frac{\ddot{y}_{\max, \text{zul}} \cdot y_{ij}}{\dot{y}_{\max, \text{zul}}^2} \right| \quad (11)$$

Es werden die folgenden, neuen dimensionslosen Größen eingeführt:

Der *Abstimmungskennwert* des Bewegungsgesetzes

$$C_\kappa = \frac{C_a}{C_v^2} \quad (12)$$

und das *Abstimmungsverhältnis*

$$\kappa = \left| \frac{\ddot{y}_{\max, \text{zul}} \cdot y_{ij}}{\dot{y}_{\max, \text{zul}}^2} \right| \quad (13)$$

zur Charakterisierung der Bewegungsgrenzen des Systems.

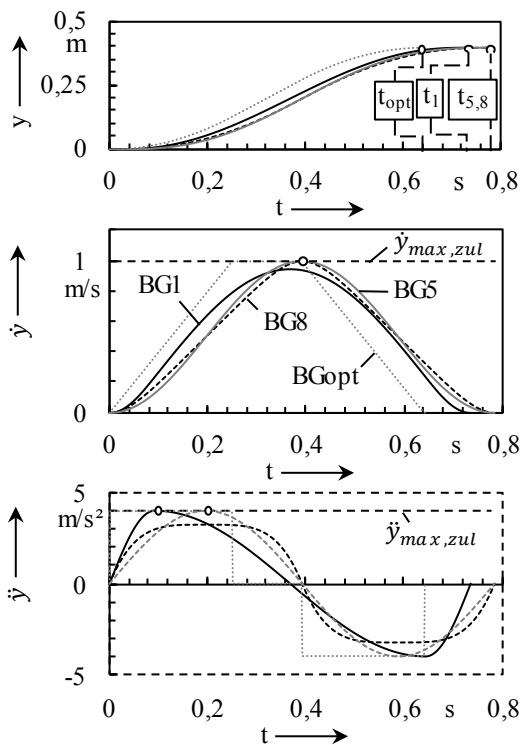


Bild 1 Fallunterscheidung

Anhand des betragsmäßigen Vergleichs dieser beiden Größen sind nachstehende allgemeingültige Bewegungscharakteristiken ableitbar (Bild 1):

Beschleunigungslimitation (Fall 1): $\kappa < C_\kappa$

Die min. zul. Taktzeit wird nach Gl. (8) bestimmt und nur die max. zul. Beschleunigung des Systems wird ausgereizt (BG1).

Fall 2: $\kappa = C_\kappa$

Die min. zul. Taktzeit wird mit Gln. (7) oder (8) bestimmt und die max. zul. Geschwindigkeit und Beschleunigung des Systems werden erreicht (BG5).

Geschwindigkeitslimitation (Fall 3): $\kappa > C_\kappa$

Die min. zul. Taktzeit wird mittels Gl. (7) bestimmt und die max. zul. Geschwindigkeit des Systems wird ausgereizt (BG8).

4 Optimierungsmethoden

Dieses Kapitel präsentiert zwei neu entwickelte Methoden zur Taktzeitoptimierung für Bewegungsabläufe in Maschinen:

Die Methode **Kennwertorientierte Funktionsauswahl (KOFA)** erleichtert die anforderungsspezifische und systemabhängige Identifikation der taktzeitbesten Funktion aus einem Katalog mit bekannten normierten Bewegungsgesetzen.

Die Methode **Kennwertorientierte Funktionssynthese (KOSY)** ermöglicht hingegen eine effiziente Anpassung einer in Parameterform vorliegenden normierten Funktion unter Berücksichtigung der Bewegungsgrenzen des Systems. Stellt der Bewegungsgesetz-Katalog keine geeignete Funktion zur Verfügung, die annähernd das gesamte Leistungspotenzial der Maschine ausnutzt, eignet sich der Einsatz dieser Methode.

4.1 KOFA

Zur Herleitung und Demonstration der Methode **KOFA** zeigt die Tabelle 1 acht normierte Bewegungsgesetze mit ihren spezifischen *Abstimmungskennwerten*, die in diesem Beitrag den zur Optimierung verfügbaren Funktionskatalog bilden. [6][8]

Tabelle 1 Funktionskatalog für Rast-in-Rast (stoß- und ruckfrei)

Bez.	Bewegungsgesetz	C_κ
BG1	Modifizierte Sinoide	1,79
BG2	Polynom 5. Grads	1,63
BG3	Sinoide von Gutman	1,28
BG4	Modifiziertes Beschleunigungstrapez	1,22
BG5	Polynom 6. Grads	1,25
BG6	Polynom 7. Grads	1,57
BG7	Polynom 8. Grads	1,17
BG8	Sinoide von Besthorn	1,57

Durch einen betragsmäßigen Vergleich der beiden neu eingeführten Größen *Abstimmungsverhältnis* und -*kennwert* sind nachstehende allgemeingültige Aussagen zur Taktzeit möglich:

Liegt beim Einsatz aller Bewegungsgesetze eine Beschleunigungslimitation (Fall 1) vor, d. h. $\kappa < \min(C_{\kappa, BG1} \cdot C_{\kappa, BG8})$, so stellt das Gesetz mit dem geringsten C_a Kennwert das Optimum dar.

Ebenso gilt für den Fall $\kappa > \max(C_{\kappa, BG1} \cdot C_{\kappa, BG8})$, d. h. es liegt in jedem Fall eine Geschwindigkeitslimitation (Fall 3) vor, dass das Gesetz mit dem betragsmäßig geringsten C_v -Kennwert das Optimum bestimmt.

Hat das vorliegende Bewegungssystem hingegen ein *Abstimmungsverhältnis*, das sich betragsmäßig im Intervall der Kennwerte befindet

$(\min(C_{\kappa, BG1} \cdot C_{\kappa, BG8}) < \kappa < \max(C_{\kappa, BG1} \cdot C_{\kappa, BG8}))$, kann aufgrund der sich auf einen kleinen Bereich beschränkten Kennwert-Beträge keine vereinfachte Aussage über die Wahl des günstigsten Bewegungsgesetzes formuliert werden. Die sich hier ergebende Taktzeit wird maßgeblich vom Funktionsverlauf und weniger von der Nähe zu den Bewegungsgrenzen, oder deren Erreichung, bestimmt. Folglich ist das Optimum nur durch einen Vergleich der zuvor einzeln ermittelten Taktzeiten identifizierbar.

Um hier eine Steigerung der Effizienz des Auswahlprozesses zu erreichen, kann unter beispielhafter Annahme der Bewegungsgrenzen des Systems

$$\ddot{y}_{\max, zul} = 1 \frac{m}{s^2} \quad (14)$$

$$\dot{y}_{\max, zul} = 1 \frac{m}{s} \quad (15)$$

bei variierendem Hub y_{ij} die äquivalenten Funktionen

$$\kappa(y_{ij}) = \left| \frac{\ddot{y}_{\max, zul} \cdot y_{ij}}{\dot{y}_{\max, zul}^2} \right| = y_{ij} \frac{1}{m} \quad \text{für } y_{ij} > 0m$$

oder

$$y_{ij}(\kappa) = \kappa m \quad \text{für } \kappa > 0 \quad (16)$$

definiert werden.

Das Einsetzen der Gln. (14) und (15) in (7) und (8) führt zur Taktzeit

$$t_{\min}^*(y_{ij}) = \begin{cases} \sqrt{y_{ij} \cdot C_a} \frac{s}{\sqrt{m}} & \text{für } \kappa < C_{\kappa} \\ y_{ij} \cdot C_v \frac{s}{m} & \text{für } \kappa \geq C_{\kappa} \end{cases}$$

jedes einzelnen Bewegungsgesetzes oder in äquivalenter Form durch ergänzendes Einsetzen der Gl. (16) zu

$$t_{\min}^*(\kappa) = \begin{cases} \sqrt{\kappa \cdot C_a} s & \text{für } \kappa < C_{\kappa} \\ \kappa \cdot C_v s & \text{für } \kappa \geq C_{\kappa} \end{cases} \quad (17).$$

Bild 2 zeigt die entsprechenden Funktionsverläufe der Bewegungsgesetze BG1...BG8.

Eine absolute Bewertung kann anhand des Beschleunigungsrechtecks BGopt (Bild 1) erfolgen. Der charakteristische Funktionsverlauf, ununterbrochen entlang einer oder beider vorgegeben Grenzen, stellt das (theoretisch erzielbare) Optimum dar, von dessen Verwendung allerdings aufgrund der bereits in der 2. Ableitung existierenden Unstetigkeiten (ruckbehaftet) abgeraten wird.

Für ein allgemeines Beschleunigungsrechteck ergibt sich die Taktzeit aus

$$t_{BGopt}(y_{ij}, \dot{y}_{\max, zul}, \ddot{y}_{\max, zul}) = \begin{cases} 2 \cdot \sqrt{\left| \frac{y_{ij}}{\ddot{y}_{\max, zul}} \right|} & \text{für } \kappa \leq 1 \\ \left| \frac{\dot{y}_{\max, zul}}{\ddot{y}_{\max, zul}} + \frac{y_{ij}}{\dot{y}_{\max, zul}} \right| & \text{für } \kappa > 1 \end{cases} \quad (18)$$

In Analogie zur Herleitung von Gl. (17) kann die optimale Taktzeit für die in Gln. (14) und (15) definierten Annahmen wie folgt berechnet werden (Bild 2):

$$t_{BGopt}^*(\kappa) = \begin{cases} 2 \cdot \sqrt{\kappa} s & \text{für } \kappa \leq 1 \\ (1 + \kappa) s & \text{für } \kappa > 1 \end{cases} \quad (19)$$

Aufgrund des proportionalen Verhaltens der resultierenden Taktzeiten für den Fall 1 (Beschleunigungslimitation, Kap. 3)

$$t_{ij, \min a} \sim \sqrt{\left| \frac{y_{ij}}{\ddot{y}_{\max, zul}} \right|}$$

und für den Fall 3 (Geschwindigkeitslimitation)

$$t_{ij, \min v} \sim \left| \frac{y_{ij}}{\dot{y}_{\max, zul}} \right|$$

können diese Diagramme bei einem beliebigen *Abstimmungsverhältnis* κ zur qualitativen Auswahl eines taktzeitgünstigen Bewegungsgesetzes hinzugezogen werden. Die zusätzlich enthaltene Information über die vorliegende Bewegungscharakteristik (Fall 1, Fall 2 oder Fall 3) unterstützt die Wahl der richtigen Gleichung, Gl. (7) oder (8), zur endgültigen Berechnung der resultierenden Taktzeit.

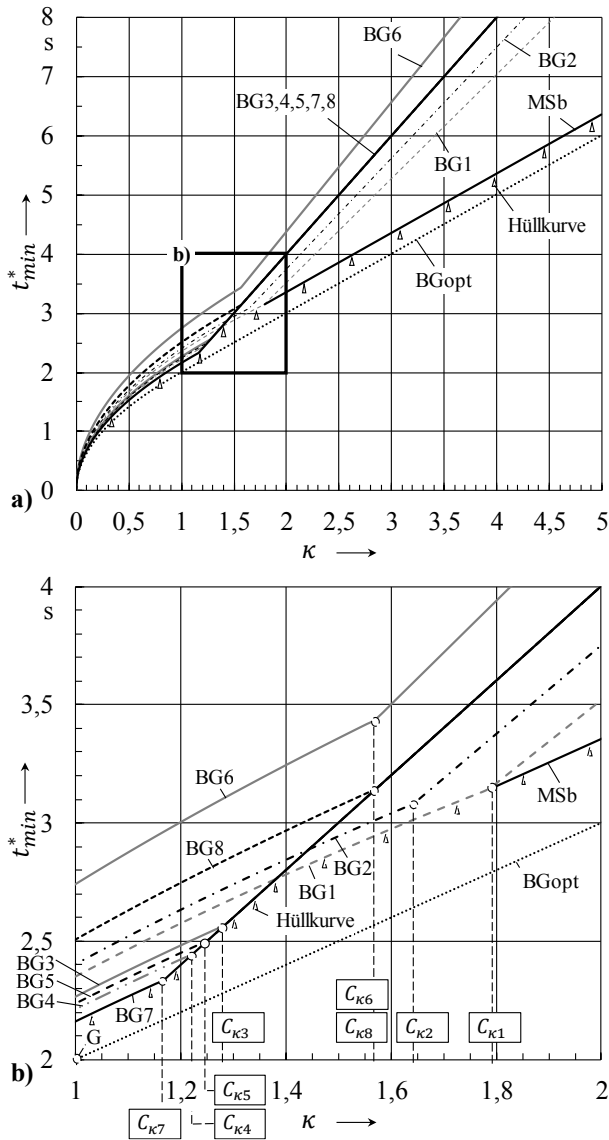


Bild 2 Taktzeiten der normierten Bewegungsgesetze a) Gesamtübersicht b) Bereichsdarstellung

Bild 3 zeigt als zusätzliche Auswahlhilfe die taktzeitbesten Bewegungsgesetze in Bezug auf das *Abstimmungsverhältnis* κ (vgl. Hüllkurve, Bild 2).

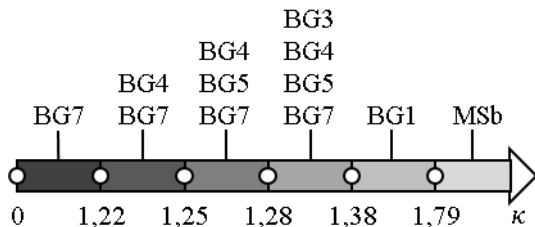


Bild 3 Auswahlhilfe

4.2 KOSY

Ein qualitativer Vergleich der im Bild 2 aufgeführten Funktionen legt dar, dass einerseits im Fall 1 (Beschleunigungslimitation, Kap. 3) mit Hilfe der bekannten Bewegungsgesetze bereits gute Annäherungen an das theoretische Optimum erzielt werden können, und andererseits im Fall 3 (Geschwindigkeitslimitation) bei steigendem *Abstimmungsverhältnis* κ ein Abdriften der Funktionen vom Optimum erkennbar ist. Das ist darauf zurückzuführen, dass vorhandene Bewegungsgesetze die max. zul. Beschleunigung des Systems immer weniger ausschöpfen.

Hierfür bietet **KOSY** einen methodischen Ansatz zur Entwicklung eines auf die Bewegungsgrenzen angepassten Bewegungsgesetzes und garantiert auch für betragsmäßig große κ eine optimale Ausnutzung des Leistungspotenzials der Maschine.

Die Beschreibung der Methode erfolgt am Beispiel der normierten Parameterfunktion *Modifizierte Sinoide mit Geradenanteil* (MSb), die aufgrund des parametrisierbaren Bereichs der Länge b mit konstanter 1. Ableitung die Geschwindigkeitsgrenze des Systems für $\kappa > 1,79$ effektiv ausnutzen kann.

Mit der Substitution

$$a(b) = 4\pi + 12\pi b - 16b + 16 \quad (20),$$

wobei b im Intervall $0 \leq b \leq 1$ definiert ist, gilt für die Funktion der MSb nach [9]:

Bereich 1: $0 \leq z < \frac{1-b}{8}$

$$f(z) = \frac{1}{a(b)} \left[4\pi z + \sin\left(\frac{4\pi z}{b-1}\right)(1-b) \right] \quad (21)$$

Bereich 2: $\frac{1-b}{8} \leq z < \frac{1-b}{2}$

$$f(z) = \frac{1}{a(b)} \left[4\pi z + 9\sin\left(\frac{4\pi z - \pi b + \pi}{3b-3}\right) \dots \right] \quad (22)$$

Bereich 3: $\frac{1-b}{2} \leq z \leq \frac{1+b}{2}$

$$f(z) = \frac{1}{a(b)} [16\pi z - 6\pi + 6\pi b - 8b + 8] \quad (23)$$

Bereich 4: $\frac{1+b}{2} < z \leq \frac{7+b}{8}$

$$f(z) = \frac{1}{a(b)} \left[4\pi z + 9\sin\left(\frac{4\pi z + \pi b - 5\pi}{3b-3}\right) \dots \right] \quad (24)$$

Bereich 5: $\frac{7+b}{8} < z \leq 1$

$$f(z) = \frac{1}{a(b)} \left[4\pi z + \sin\left(\frac{4\pi - 4\pi z}{b-1}\right) \dots \right. \\ \left. \dots (1-b) - 16b + 12\pi b + 16 \right] \quad (25)$$

Für die Kennwerte gelten die Gleichungen:

$$C_v(b) = \frac{4\pi}{\pi - 4b + 3\pi b + 4} \quad (26)$$

$$C_a(b) = -\frac{4\pi^2}{(b-1)(\pi - 4b + 3\pi b + 4)} \quad (27)$$

Durch Einsetzen der Gln. (26) und (27) in Gl. (12), gilt für den *Abstimmungskennwert*:

$$C_\kappa(b) = \frac{C_a(b)}{C_v^2(b)} = -\frac{\pi - 4b + 3\pi b + 4}{4(b-1)} \quad (28)$$

Für die Parametrierung der *MSb* wird zunächst Gl. (28) in die nachstehende äquivalente Form gebracht:

$$b = -\frac{\pi - 4C_\kappa + 4}{4C_\kappa + 3\pi - 4} \quad (29)$$

Ein Bewegungsgesetz, dass die Grenzen des Systems ausreizt, fordert die Bedingung

$$C_\kappa = \kappa \quad (\text{Fall 2}).$$

Somit ist der Parameter b anhand des *Abstimmungsverhältnisses* κ gemäß Gl. (30) ermittelbar.

$$b = -\frac{\pi - 4\kappa + 4}{4\kappa + 3\pi - 4} \quad (30)$$

Das synthetisierte Bewegungsgesetz ergibt sich folglich nach Gln. (21) bis (25).

Zusammenfassend kann gesagt werden, dass sich Bewegungsgesetze mit Hilfe dieser neu entwickelten Vorgehensweise einfach und effizient synthetisieren lassen, und dass dadurch, besonders für betragsmäßig große κ , im Vergleich zu bekannten Gesetzen, deutliche Taktzeit-Verbesserungen erzielbar sind (Bild 2).

5 Beispiel zur Anwendung der Methoden

Zur Realisierung nacheinander folgender Umsetzbewegungen von in der Masse variierenden Produkten innerhalb einer Handhabungsstation, kommt aufgrund der Produkt-Sensibilität ein mit geringem Anpressdruck betriebener Sauggreifer zum Einsatz (Bild 4). Die Bewegung wird mittels eines Motion-Control-Systems erzeugt und kann im Betrieb beliebig angepasst werden.

Gefordert ist eine Optimierung des Zeitfensters t_{ij} für den Umsetzprozess. Bild 5 zeigt dazu die entsprechende Bewegungsaufgabe.

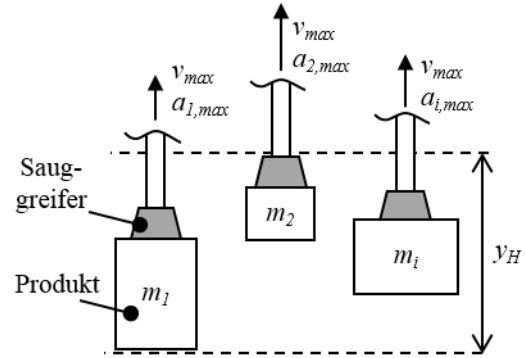


Bild 4 Anwendungsfall

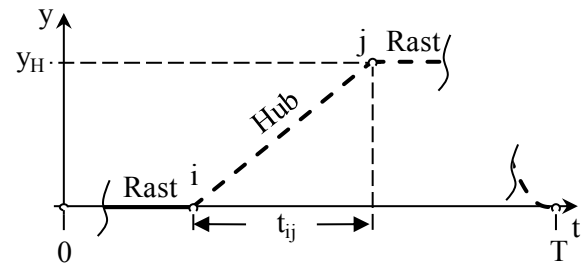


Bild 5: Umsetzbewegung

Prozessbedingt und unter Beachtung der Mechanik gibt es folgende Restriktionen:

- Hub $y_H = 0,3\text{m}$
- max. zul. Geschwindigkeit $v_{\max} = 3 \frac{\text{m}}{\text{s}}$

Die vorhandene Saugkraft des Greifers erlaubt unter Beachtung der wirkenden Gewichts- sowie Trägheitskräfte max. Beschleunigungen von:

$$a_{1,\max} = 40 \frac{\text{m}}{\text{s}^2} \quad (\text{Produkt 1})$$

$$a_{2,\max} = 75 \frac{\text{m}}{\text{s}^2} \quad (\text{Produkt 2})$$

Die Anwendung der entwickelten Methoden zur Optimierung des geforderten Prozesses wird im Folgenden am Beispiel der Produkte 1 und 2 demonstriert:

Prozessoptimierung für Produkt 1 (P_1):

-Schritt 1-

Berechnung des Abstimmungsverhältnisses nach Gl. (13)

$$\kappa_1 = 1,33$$

-Schritt 2-

Anwendung von KOFA:

Der Anwender hat die Wahl zwischen den Gesetzen BG4, BG5, und BG7 (Bild 3), die angesichts der betragsmäßig gleichen Geschwindigkeitskennwerte $C_v = C_{v,BG4} = C_{v,BG5} = C_{v,BG7} = 2$ und dem vorliegenden Fall 3 (s. Kap. 3) Taktzeiten von jeweils

$$t_{12} = \frac{y_H}{v_{\max}} \cdot C_V = 0,2 \text{ s}$$

erzielen.

Gegenüber dem häufig verwendeten Polynom 5. Grads wird eine Verbesserung um ca. 4% erzielt ($t_{12,BG2} = 2,08 \text{ s}$). Welches der drei Bewegungsgesetze schließlich in Frage kommt, kann anhand weiterer Kriterien entschieden werden. In diesem Fall wurde das Bewegungsgesetz BG4 ausgewählt (Bild 5).

Prozessoptimierung für Produkt 2 (P₂):

-Schritt 1-

Berechnung des Abstimmungsverhältnisses nach Gl. (13)

$$\kappa_2 = 2,5$$

-Schritt 2-

Anwendung von KOSY:

In diesem Fall wird nach Bild 3 die Anpassung der parametrischen Funktion MSb empfohlen.

Nach Gl. (30) folgt

$$b = 0,185$$

Für die Taktzeit resultiert:

$$t_{12} = \frac{y_H}{v_{\max}} \cdot C_V = 0,154 \text{ s}$$

$$\text{mit } C_V(b = 0,185) = 1,54 \text{ (nach Gl. (26))}$$

Im Vergleich zum Polynom 5. Grads wird hier eine Verbesserung um ca. 18% erzielt ($t_{12,BG2} = 1,88 \text{ s}$).

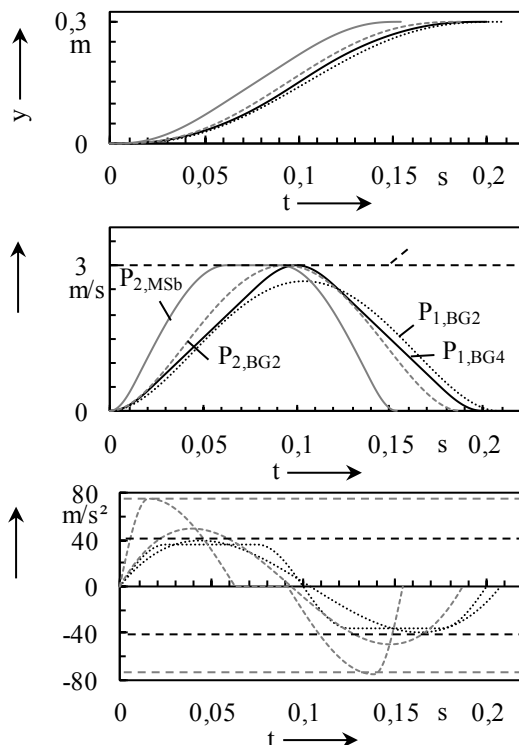


Bild 5 Optimierungsergebnis des Anwendungsfalls

6 Zusammenfassung und Ausblick

In diesem Beitrag wird eine Vorgehensweise zur Optimierung von Bewegungen hinsichtlich der Taktzeit beschrieben. Die beiden neu entwickelten Methoden KOFA und KOSY ermöglichen die effiziente Auswahl und Synthese von Bewegungsgesetzen, die nahezu das gesamte Leistungsvermögen der Maschine zur Steigerung der Taktzeit nutzen.

Die geringe Komplexität der Ansätze im Vergleich zu anderen Optimierungsverfahren erlaubt es, dass Anwender ohne Expertenwissen gute Ergebnisse erzielen. Des Weiteren erfüllt der Ansatz die notwendige Voraussetzung zur softwareseitigen Implementation in die Steuerung, um im Betrieb Online-Profil-Anpassungen vornehmen zu können. Durch die Online-Berechnung produktspezifischer taktzeitoptimierter Bewegungsprofile wird auch im Kontext der im industriellen Umfeld geprägten Ausrichtung nach Flexibilität und Industrie 4.0 eine effiziente Methode zur Optimierung von Bewegungsbereichen bis hin zu gesamten Technologiefunktionen ermöglicht.

In Zukunft wird sich der Autor mit der Erweiterung des Ansatzes für andere Bewegungsaufgaben, wie G-G oder G-U, sowie mit der Entwicklung anderer Bewegungsgesetze befassen. Die Erprobung der Gesetze hinsichtlich dynamischer Effekte mit Hilfe eines vorhandenen Prüfstands ist ebenso geplant.

7 Literatur

- [1] ALPERS, Burkhard: Zur Vermeidung des kinematischen Schwingens beim Bewegungsdesign. In: VDI (Hrsg.). Bewegungstechnik. Düsseldorf: VDI-Verl., 2014 (VDI-Berichte, 2237), S. 51–62
- [2] ALPERS, Burkhard: Schranken und Extremalfunktionen für die Kennwerte der VDI Richtlinie 2143. In: VDI (Hrsg.). Bewegungstechnik. Düsseldorf: VDI-Verlag, 2016 (VDI-Berichte, 2286).
- [3] BRAUNE, Reinhard: Bewegungs-Design mit interaktiver Variation von Stützpunkten. In: VDI (Hrsg.). Bewegungstechnik. Düsseldorf: VDI-Verl., 2006 (VDI-Berichte, Nr. 1966), S. 47–76
- [4] HEINE, Andreas; MATTHES, Jörg; SCHULZ, Ringo; BERGER, Maik: MOCAD – Ein grafisch interaktives Werkzeug zur Berechnung und Optimierung von Kurvengetrieben und Motion Control Systemen. In: VDI (Hrsg.). Bewegungstechnik. Düsseldorf: VDI, 2010 (VDI-Berichte, 2116).
- [5] PLAB, Wilfried L.: Online Auslegung von Kurvenscheiben. In: VDI (Hrsg.). Bewegungstechnik. Düsseldorf: VDI, 2010 (VDI-Berichte, 2116), S. 87–96
- [6] VDI Richtlinie 2143 Blatt 1. 1980. Bewegungsgesetze für Kurvengetriebe
- [7] FRICKE, Andreas; GÜNZEL, Detlef; SCHAEFFER, Thomas: Bewegungstechnik: Konzipieren und Auslegen von mechanischen Getrieben. München: Fachbuchverl. Leipzig im Hanser-Verl., 2015
- [8] VOLMER, Johannes (Hrsg.); MÜLLER, Jörg (Hrsg.): Getriebetechnik: Kurvengetriebe. 2., stark bearb. Aufl. Ilmenau, Berlin: Ges. zur Förderung der Digitalen Mechanismen- und Getriebebibliothek; Verl. Technik, 1989
- [9] HEINE, Andreas: Ein Beitrag zur kennwertorientierten Entwicklung kurvengesteuerter, ebener Schrittgetriebe. Chemnitz, Universitätsbibliothek, TU Chemnitz. Dissertation. 2015

Design und Konstruktion eines neuen Konzepts für Mehrteilgreifer

Liang Chen, Philipp Wabnitz, Prof. Dr.-Ing. Maik Berger, Fan Yin
Professur Montage- und Handhabungstechnik
Technische Universität Chemnitz, 09126 Chemnitz, Deutschland
liang.chen@s2013.tu-chemnitz.de; philipp.wabnitz@mb.tu-chemnitz.de

Kurzfassung

Im ersten Teil des Beitrags soll das neue methodische Endeffektorsystem bzw. Greifsystem vorgestellt werden. Basierend auf den Begriffen „Handhaben“ und „Greifer“ aus den VDI Richtlinien 2740 werden jetzige Maschinen allgemein neu definiert und entsprechend in drei Stufen klassifiziert. Dort werden alle industriellen Maschinen nicht als Roboter oder Pick&Place-Geräte, sondern nur als eine Greifzelle betrachtet. Jede Greifzelle kann dabei theoretisch beliebig mit anderen kombiniert werden, jedoch sind nicht alle Kombinationen sinnvoll. Der Fokus des Beitrags liegt nun in der Analyse günstiger Kombinationen zur Entwicklung neuer Greifkonzepte.

Im zweiten Teil des Beitrags soll das Konzept eines neuartigen Greifers (abgeleitet vom o. g. Prinzip) für unterschiedliche Werkstückgeometrien bzw. Greifaufgaben vorgestellt werden. Die Drehfunktion wird dabei durch ein Schrittgetriebe realisiert. Alle Einzelteile des Prototyps werden mittels 3D-Druck hergestellt und sind somit leichter und günstiger als herkömmliche Greifer. Der Antrieb erfolgt über zwei Servomotoren. Der Mehrteilgreifer wiegt weniger als 1 kg bei einem Hub von 20 mm pro Backe und ermöglicht es vier verschiedene Objekte zu greifen.

Abstract

In the first part of the offered article a novel methodology of end effector system or gripping system will be introduced. Based on the definitions of “Handling” and “Gripper” from the VDI guidelines 2740 [1], the common machines are re-defined and correspondingly classified in three levels. Then all the industrial machines are not regarded as robots or pick-and-place devices but only as a gripping cell individually. Theoretically each gripping cell can be optionally combined with others, however, not all the combinations are meaningful. This article focuses on the significant combinations for developing the new concept of gripping system.

In the second part of this article a novel gripper which is derived from the aforementioned principle, for different geometric workpieces or gripping tasks will be presented. The driving system is realized by two servomotors, one of which is rotated through intermittent mechanism. All parts of the prototype are produced by 3D printing technology and therefore lighter and cheaper than conventional grippers. The multifunctional gripper weighs less than 1 kg with a stroke of 20 mm per finger and enables four different objects.

1 Einleitung

Intelligent, kollaborativ, flexibel und effizient, das sind die meist verwendeten Begriffe zur Beschreibung einer modernen ‚Smart Factory‘. Demnach gehören auch konventionelle Arbeitsstrukturen, welche Menschen und automatisierte Maschinen strikt voneinander trennen, der Vergangenheit an. Denn eine Kollaboration zwischen Mensch und Maschine nutzt die Vorteile beider Seiten. Die Flexibilität und Intuition eines Menschen ergänzt dabei die Effizienz und Genauigkeit der Maschinen. Ein repräsentatives Szenario dafür ist die Mensch-Roboter-Kollaboration, auf deren Grundlage Menschen neben und zusammen mit diesen kollaborierenden Robotern (englische Abkürzung: *cobot*) sicher und problemlos arbeiten. Die *cobots* erfüllen dabei ihre Aufgaben im alltäglichen Einsatz gut bis sehr gut, jedoch nur ohne Berücksichtigung der eingesetzten Endeffektoren, da diese meist keinen kollaborierenden Betrieb erlauben. Durch die Minimierung der Abmessungen und Gewichte von Industrieprodukten wie Robotern, Handhabungs- und Montagevorrichtungen, elektrischen Geräten und Werkzeugen wird

eine Standardisierung und Modularisierung ermöglicht. Im folgenden Kapitel werden die üblichen Industriemaschinen in drei Stufen eingeteilt, und die sinnvollen Kombinationen jeder Ebene werden ausführlich analysiert.

Um die Leistungsfähigkeit von Montage- oder Handhabungsprozessen zu erhöhen, werden vielfältige Greifsysteme, z.B. Mehrfingergreifer, Greifer-Magazine oder Revolver, verwendet. Besonders im Bereich der kollaborierenden Robotik spielen jedoch flexible, leichte und multifunktionale Greifer eine zunehmend wichtigere Rolle. Die derzeit am Markt befindlichen Mehrteilgreifer- oder Revolversysteme besitzen meist ein hohes Gewicht bei gleichzeitig großen Abmessungen und sind aus diesem Grund nicht für Mensch-Roboter-Kollaboration geeignet. Statt einer Revolverbaugruppe mit mehreren Greifern sind die Greifbacken des neuen Mehrteilgreifers drehbar ausgeführt und somit umschaltbar für die jeweilige Aufgabe. In Kapitel drei wird das Konstruktionsdesign dieses Greifers vorgestellt und anschließend optimiert.

2 Die Greiferzellenpyramide

Laut Hesse (2011), bezieht sich die Greiferflexibilität auf die Eigenschaft eines Greifers, insbesondere Werkstücke mit sehr unterschiedlichen Formen und Abmessungen ohne Umbau anfassen zu können. [2] Bild 1 zeigt vier Varianten der Greiferflexibilität: Werkstückform, Größe, Masse und Zusatzfunktionen.

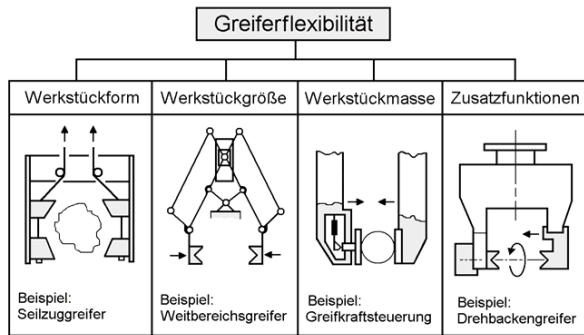


Bild 1 Merkmale flexibler Greifer [2]

Flexible Greiftechniken haben insbesondere bei komplexen Montage- und Handhabungsaufgaben eine große Bedeutung, bei denen Einzelteile mit unterschiedlicher Größe, Form, Masse und Material durch einen Roboter gegriffen und montiert werden. Normalerweise kann dies durch folgende Methoden realisiert werden:

- *Wechselsysteme, die spezialisierte Einzelgreifer nacheinander zum Einsatz bringen*
- *Mehrfachgreifer als kompakte Einheit, wie z.B. Revolvergreifer*
- *Mehrstellengreifer, dessen Fingerglieder sich an verschiedene Objektkonturen anschmiegen*
- *Mehrgliedrige Gelenkfingergreifer, dessen Fingerglieder sich an verschiedene Objektkonturen anschmiegen.*

[2]

Basierend auf den obigen Definitionen kann im Folgenden ein neuartiges Greifsystem mit der Bezeichnung „Greiferzellenpyramide“ abgeleitet werden.

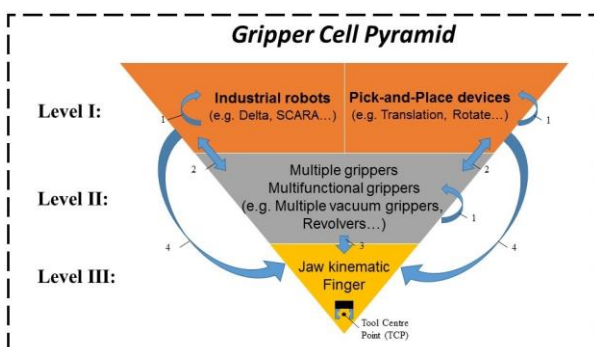


Bild 2 Greiferzellenpyramide

Definitionen:

Level I: Ein Basisterminal, das direkt am Boden montiert werden kann und über mehrere Bewegungsfreiheitsgrade verfügt, wie alle typischen Industrieroboter und Pick & Place Geräte mit translatorischen oder rotatorischen Achsen.

Level II: Eine Zwischenverbindung zwischen Level I und Level III, die die Handhabungskapazität durch „Mehrfachgreifer“ (gleichzeitig [1]) oder „Revolvergreifer“ (zeitlich und funktionell unabhängig [1]) erhöht. Diese Stufe ist optional für das gesamte Greifsystem, bietet jedoch immense Flexibilität für Roboter und erhebliche Effizienz innerhalb der Bearbeitungszykluszeit.

Level III: Ein Werkzeugterminal, das in einem Greifsystem einfach an der Endposition positioniert wird. Eine Seite ist mit der Schnittstelle einer höheren Ebene verbunden und die andere muss ein Tool-Centre-Point (TCP) sein, z.B. Backen und Finger

2.1 Die Symbologie des Greifsystems

Theoretisch können die Greifeinheiten beliebig und unendlich kombiniert werden, die meisten dieser komplexen Kombinationen sind jedoch bedeutungslos. Daher ist ein offensichtlicher und genauer Ausdruck vorteilhaft für das Verständnis, die Klassifizierung und die Ableitung des komplexen Greifsystems. Zunächst werden die Schreibsymbole definiert, die zur Darstellung eines Greifsystems verwendet werden.

‘I, II, III’, bezeichnen die Greifeinheiten aus jeder Stufe.

‘/’, bezeichnet eine zusammengesetzte (parallele) Kombination mit zwei Greifeinheiten.

‘-’, bezeichnet eine (serielle) Standardkombination mit zwei oder mehr Greifeinheiten.

‘//’, unterscheidet die verschiedenen Greifeinheiten in einer niedrigeren Klasse, wenn in der oberen Klasse „/“ erscheint.

Zusammen mit den Schreibsymbolen können grafische Symbole das System explizit veranschaulichen.

Level I wird als ‘I’ geschrieben und als Basistyp und Einheitstyp gezeichnet.

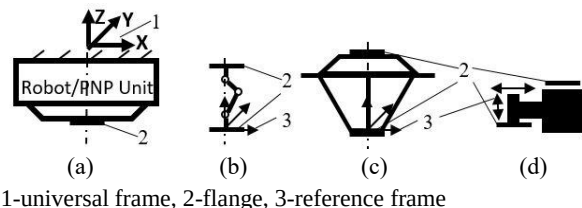


Bild 3 I: Industrieroboter + PNP Geräte

(a) Allgemeiner Basistyp von Level I

(b) Einheitstyp der Serienkinematik

(c) Einheitstyp der Parallelkinematik

(d) Einheitstyp der Pick & Place Geräte.

Ein Greifsystem muss zunächst mit einem allgemeinen Basistypen (Bild 3a) als ersten Ausgangspunkt gekennzeichnet sein. Dieses Zeichen wird nicht nur durch ein Gestell, sondern auch durch den universellen Koordinaten

tenrahmen gebildet. (1-universal frame) Im Gegensatz zum Universal-Frame bezeichnet 3 einen Referenz-Frame und damit (b), (c) und (d) den Einheitstyp von serieller Manipulatoren, paralleler Manipulatoren und Pick & Place Geräten. 2 ist eine Standardschnittstelle, die allgemein als Flansch bezeichnet wird.

Level II wird als „II“ geschrieben und bildlich als passiver Typ und aktiver Typ dargestellt.

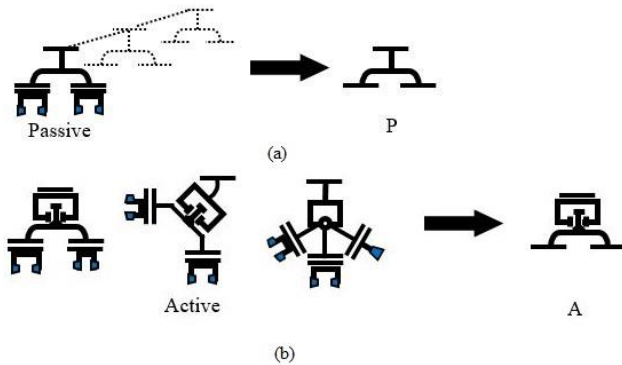


Bild 4 II: Mehrfachgreifer + Revolvergreifer
(a) Mehrfachgreifer (passive Betätigung)
(b) Revolvergreifer (aktive Betätigung)

Bild 4 zeigt die typischen Doppelgreifer mit zwei Schnittstellen. Die können möglicherweise als gestrichelte Linie auf mehrere Greifer erweitert werden. Diese Art wird aus Mangel an Stellgliedern als „P“ bezeichnet (kurz für Passiv) und als rechte Seite des Pfeils vereinfacht. In gleicher Weise sind die Revolvergreifer aktive Einheiten und werden in einem Symbol (Bild 4b) zusammengefasst. Das grafische Symbol des Revolvers ist für fast alle Fälle geeignet, obwohl sie normalerweise komplizierter und mit mehr Schnittstellen gestaltet sind.

Level III wird als „III“ geschrieben und ebenfalls mit einem Symbol (Bild 5) dargestellt.



Bild 5 III: Backenkinematik, Finger oder Werkzeug

Um bestimmte Aufgaben zu erfüllen, werden wahrscheinlich die Basiseinheiten aus Level II und III integriert um neue Varianten abzuleiten, die als „Integrated Units (IU)“ gruppiert und benannt werden. Aufgrund der möglichen Redundanz des Freiheitsgrads und Singularität werden die integrierten Einheiten von Level I nicht diskutiert. Im Allgemeinen sollte eine sinnvolle Integration nur einmal erfolgen, ansonsten bringt sie eine ähnliche Komplexität wie die serielle Kinematik und kann durch andere optimale Methoden ersetzt werden. Offenbar gibt es nur drei Möglichkeiten durch die Permutationen und Kombinationen von II und III: II mit II, III mit III sowie II mit III.

Integrierte Einheiten II mit II werden als „IIII“ geschrieben, als drei Arten dargestellt: aktiv + aktiv (AA), aktiv + passiv (AP) und passiv + aktiv (PA). Die Art von passiv + passiv ist eigentlich gleich dem passiven Level II (Bild

4a). Alle drei Grafiksymbbole können zufällig angewendet werden und gehören einheitlich zum Level II.

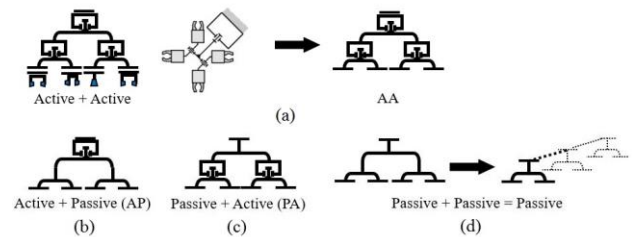


Bild 6 IU Level III

Integrierte Einheiten III mit III werden als „IIIIII“, die zum Level III gehören, wobei es vier Typen gibt: parallel + parallel (PP), angular + angular (AA), parallel + angular (PA), angular + parallel (AP). Die Grafiksymbbole sind frei verwendbar und können auch als PP Typ dargestellt werden. Mit einem weiteren Stellantrieb in den oberen Klassen haben IUs von IIIIIII die Fähigkeit, den Hub zu vergrößern. (Bild 7)

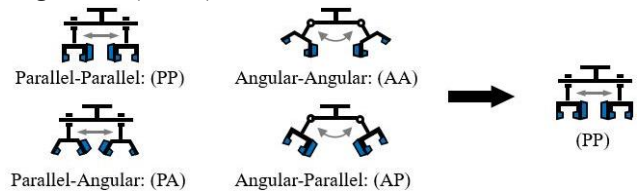


Bild 7 IU Level IIIII

Inspiziert von IU Level IIIII und IIIIIII sollte es die dritte integrierte Einheit geben, die als „IIIIII“ bezeichnet wird. Jedoch abweichend von den herkömmlichen Anwendungsfällen: Level II um Level III angehängt (Bild 4), muss IU Level IIIII integriert sein, damit es als Einheit kompakt funktioniert.

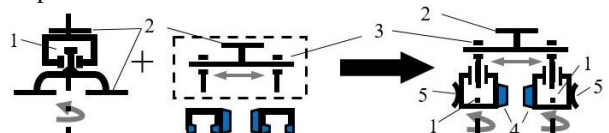


Bild 8 IU Level IIIII

IU Level IIIII wird als einheitliches Symbol dargestellt (Bild 8). Parallel werden zwei Drehachsen der Greifachse zugeordnet, durch die 4~6 Fingerpaare für unterschiedliche geometrische Objekte verändert werden können. Aufgrund der unlösbaren Struktur zwischen Level II und Level III funktionieren die integrierten Einheiten von Level IIIII als vollständiger Endeffektor und werden daher in Level III klassifiziert.

2.2 Kombinationsmethoden: Zusammengesetzt und Standard

Bis hier hin wurden die Grafik- und Schriftsymbole aller Einheiten inklusive Basis und Integrierte aufgezeigt. Sie sind Level I, Level II, IU Level IIIII, Level III, IU Level IIIIIII und IU Level IIIIIII. Die Beziehungen dieser sechs Level Einheiten sind in Bild 9 dargestellt. Unter Verwendung der Symbole „/“ und „-“ werden die Kombinations-

methoden einfach in zusammengesetzte Typen bzw. parallele Konstruktionen und Standardtypen bzw. serielle Strukturen unterteilt.

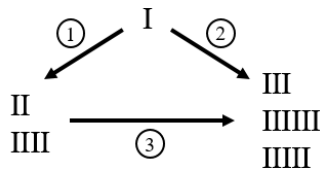


Bild 9 Beziehungen von sechs Level Einheiten

- Kombinationsmethode des zusammengesetzten Typs mit dem Schriftsymbol „/“

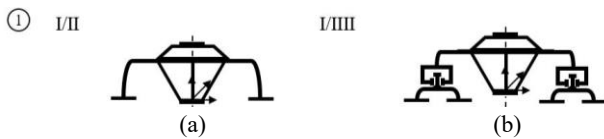


Bild 10 Zusammengesetzte Typen von Level I und II
(a) I/II Komposition
(b) I/IIII Komposition

Hier können die Einheiten von Level I also serielle Manipulatoren, parallele Manipulatoren oder Pick & Place Geräte genauso wie bei den anderen Level kombiniert werden. Bild 10a zeigt den zusammengesetzten Typ zwischen parallelem Manipulator aus Level I und der passiven Einheit aus Level II. Bild 10b veranschaulicht den zusammengesetzten Typ zwischen parallelem Manipulator aus Level I und PA-Einheit aus Level IIII (Bild 6c).

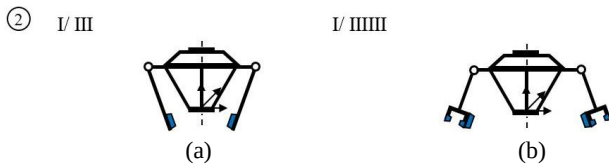


Bild 11 Zusammengesetzte Typen von Level I und III
(a) I/III Komposition
(b) I/IIII Komposition

Ähnlich wie bei den Kompositionen der Gruppe ① zielen die Einheiten, die aus Level I und III zusammengesetzt werden, darauf ab, die Greifobjekte zu diversifizieren. Eine bemerkenswerte Eigenschaft von Gruppe ① und ② ist, dass die Einheit aus Level I durch die parallele Konfiguration eine kollaborative Unteraufgabe ausführen kann, wodurch die Betriebszeit äußerst verkürzt werden kann. Aufgrund des ausgeklügelten Aufbaus werden IUs aus Level IIIII vom zusammengesetzten Typ ausgeschlossen. Im Gegensatz zu den beiden vorherigen Gruppen scheint der letzte zusammengesetzte Typ ③ nicht sinnvoll genug zu sein. Ein Hauptgrund ist, dass das Level II eine ähnliche Wirkung wie parallele Kompositionen für die Flexibilität der Greifer hat. Außerdem fehlt es beiden an Freiheitsgraden, weshalb die meisten Kompositionen nicht unersetzlich sind (genauso wie bei Level IIII).

- Kombinationsmethode des Standardtyps mit dem Schriftsymbol „-“

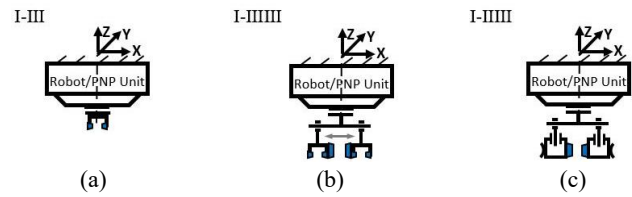


Bild 12 Standardtypen von Level I und III
(a) I/III Standardverbindung
(b) I/IIII Standardverbindung
(c) I/IIII Standardverbindung

Das klare Merkmal des Standardtyps ist die Flanschkuppelung, die strukturell wie die serielle Kinematik ist. Bild 12a zeigt die gängigste Standardverbindung mit Roboter und Greifer (I-III). Offensichtlich kann diese einfache Kombination die flexiblen Roboter nicht voll ausnutzen. Daher werden die neuartigen multifunktionalen Greifer mit einem weiteren Freiheitsgrad wie (b) oder (c) entwickelt.

2.3 Anwendungsfälle des hierarchischen und modularen Greifsystems

Durch die Standardverbindung mit dem Symbol „-“ lassen sich die Greifeinheiten als hierarchisches System leicht analysieren. Beispielsweise können die Kombination in Bild 12 als zweistufiges Greifsystem I-III oder I-IIII definiert werden, jedoch ist die Verwendung dieser Symbole für eine einfache Konstruktion eher umständlich. Dennoch, mit der Realisierung der redundanten Verbindung von zwei oder mehr Robotern, der Entwicklung hin zur intelligenten Industriefabrik sowie folglich mehr Hierarchien des Greifsystems, werden sogar schon dreistufige Systeme zu einer Herausforderung. Wie bereits erwähnt, ist die Hierarchisierung und Modularisierung des Greifsystems vorteilhaft, um neue Strukturgreifer abzuleiten oder zu nutzen.

Derzeit sind fünf oder mehr Stufen noch nicht für die Forschung interessant, daher werden drei- und einige vierstufige Systeme detailliert beschrieben.

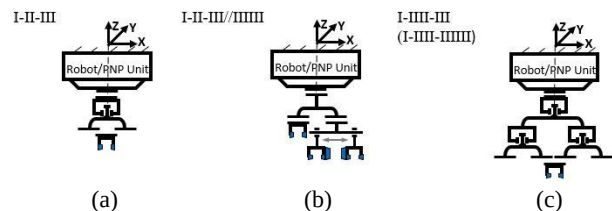


Bild 13 Normale dreistufige Greifsysteme

Bild 13a zeigt ein typisches dreistufiges Symbol eines normalen industriellen Falls, bei dem es sich um einen Roboter mit Revolver plus Greifern handelt. Wenn sich einer der Endeffektoren von anderen Einheiten unterscheidet, kann er als Bild 13b symbolisiert werden. Level III und IU Level IIIII müssen jedoch durch „/“ getrennt werden, was bedeutet, dass diese beiden Einheiten sich

(möglicherweise mehr) auf derselben Klasse befinden. Es ist zu beachten, dass die aus drei verschiedenen Einheiten aufgebaute Schnittstelle der Multi-Greifer ein Sonderfall ist, d.h. der Fall „III//IIII//IIII“ ist bedeutungslos. Dies ist ein Grund, warum für das grafische Symbol von Level II nur zwei Schnittstellen ausreichen. Bild 13c stellt eine andere Variante mit Level III in der zweiten Klasse dar. Ein wichtiger Punkt ist, dass III nicht mit II-II ersetzt werden kann, was in der Praxis definitiv keine Bedeutung hat. Da funktioniert die integrierte Einheit aus Level III nicht nur mechanisch, sondern auch elektrisch als Ganzes.

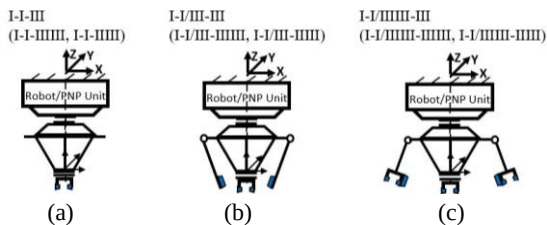


Bild 14 Dreistufiges Greifsystem mit zusammengesetzten Typ von Level I und III

Verglichen mit den dreistufigen Anwendungen in Bild 13, die bereits in der Industrie Verwendung finden, befinden sich die neuartigen dreistufigen Greifsysteme in Bild 14 zum größten Teil noch in der Forschung. Als Darstellung (a) haben kürzlich Wissenschaftler die Probleme serieller und paralleler Hybridmanipulatoren untersucht. [4] [5] [6] Darüber hinaus zeigen (b) und (c) zwei Varianten, deren zweite Klasse zusammengesetzte Typen von Level I und III.

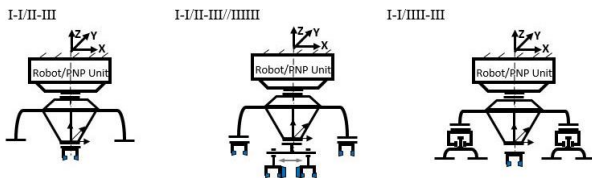


Bild 15 Dreistufiges Greifsystem mit zusammengesetzten Typ von Level I und II

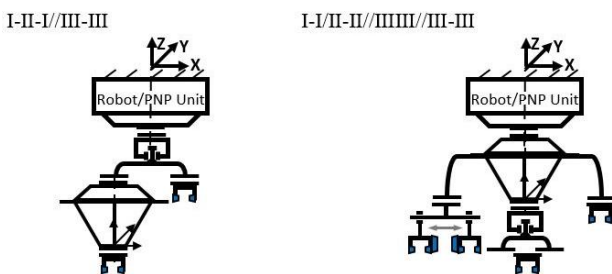


Bild 16 Vierstufiges Greifsystem

Die letzte Möglichkeit eines dreistufigen Systems und vierstufigen Systems mit zusammengesetztem Typ werden in Bild 15 und Bild 16 symbolisiert. Allerdings muss eine solche komplexe Konstruktion mit daraus folgenden Kollisionsproblemen in Zukunft diskutiert werden. Abschließend werden die Prinzipien der Symbologie des Greifsystems zusammengefasst.

- 1) Sowohl in Schrift- und als auch grafischer Symbologie muss das System mit Level I gestartet werden.
- 2) Sowohl in Schrift- und als auch grafische Symbologie muss das System mit Level III beendet werden.
- 3) Das Symbol „/“ wird für parallele Kombinationen verwendet, die als Modul zusammengesetzt sind. Die zusammengesetzten Einheiten können mit „/“ getrennt werden, unterscheiden sich jedoch bei Standardtypen.
- 4) Das Symbol „-“ wird für Serienkombinationen durch Standardflansche verwendet.
- 5) Wenn im System ein „/“ verwendet wurde, kann dies als geteilte Schaltung verstanden werden. Wie in Bild 17a dargestellt, wird der obere Zweig von I, gefolgt von einem weiteren I in der zweiten Klasse und dann mit III abgeschlossen. Der untere Zweig wird von der gleichen Einheit aus Level I gestartet, aber von der anderen Einheit aus Level III beendet. Da gibt es keine Bedeutung für III-III.
- 6) Das Symbol „/“ wird für die Unterscheidung der Einheiten in der unteren Klasse verwendet. Zum Beispiel in Bild 17b, wird die dritte Klasse mit II//IIII//III kombiniert. Da Level I nur eine Schnittstelle hat, ist das Level II in der dritten Klasse entsprechend mit Level I in der zweiten Klassen verbunden. Dann gehören die beiden anderen Einheiten (IIII//III) definitiv zum Level II in der zweiten Klasse. Daher spielt die Reihenfolge der Einheiten eine bedeutende Rolle.



Bild 17 Prinzip der Symbologie des Greifsystems

3 Prototyp des Mehrteilgreifer aus IU Level IIIII

Nach der Aussage von Hesse, gibt es zwei Entwicklungsrichtungen für die Flexibilisierung des Greifers.

- Suche nach der einfachsten Struktur, mit der möglichst viele flexible Funktionen erreicht werden können. Solche Entwicklungen zielen auf eine industrielle Verwendung. [2]
- Design nach dem Vorbild der menschlichen Hand, was sehr komplexe Roboterhände zur Folge hat. Die Entwicklungen zielen auf eine Verwendung in der Servicerobotik und Weltraumtechnik. [2]

Der in diesem Beitrag neu entwickelte multifunktionale Greifer konzentriert sich auf die industrielle Verwendung. Eigentlich ist die Struktur dieses Greifers sehr ähnlich dem Mini-Revolver oder einer Variante eines Drehbackengreifers, der um die Z-Achse rotiert (in Bild 1 ist ein um die X-Achse rotierender Drehbackengreifer zu sehen). Um die Fingerrotation mit einer einfachen Struktur zu realisieren,

wird ein halbes Malteserkreuz¹ angewendet wie in Bild 18 dargestellt.

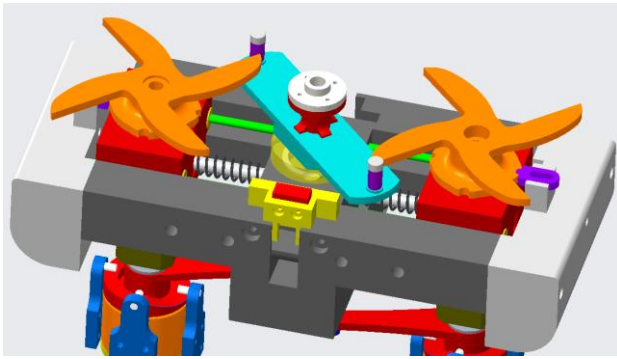


Bild 18 Halbes Malteserkreuz

Der Hauptgrund für die Verwendung des Malteserkreuzes ist die große Toleranz, die den Nachteil der fehlenden Positionsregelung der Servomotoren ausgleichen kann. Demgegenüber stehen die Vorteile des PDI-6221MG Servomotors: sein geringes Gewicht bei großem Drehmoment von 20 kg.cm und die Fähigkeit um 360 Grad unbegrenzt drehen zu können. Wie in Tabelle 1 gezeigt, sind die Gesamtkosten aller Komponenten geringer als 120 €.

Komponenten	Preis	Gewicht
Servo Brick	49,99 €	18 g
IO-16 Bricklet	8,99 €	30 g
2×Servomotor	<2×10 €	2×62 g
Gleitlager	< 20 €	
PLA Material	< 20 €	
Summe	< 120 €	< 1 kg

Tabelle 1 Preisliste der wichtigen Komponenten

Um die Kollision zwischen den Greif- und Drehbewegungen zu vermeiden, ist eine Nut so gestaltet, dass sich der Finger während des Greifens nicht automatisch drehen kann (Bild 19a). Außerdem wird die Positionierkonstruktion des Malteserkreuzes von innen nach außen verschoben (Bild 19b).

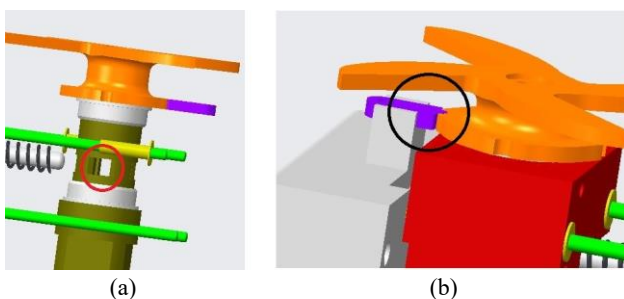


Bild 19 Interne Struktur des Mehrteilgreifers

Ein endgültiger echter Prototyp wird in Bild 20 dargestellt. Dieser ermöglicht es Objekte mit vier verschiedenen Geometrien zu greifen: die Schmalseite eines Radiergummis,

die Breiteseite eines Radiergummis, ein hohles Rohr und einen Legostein.

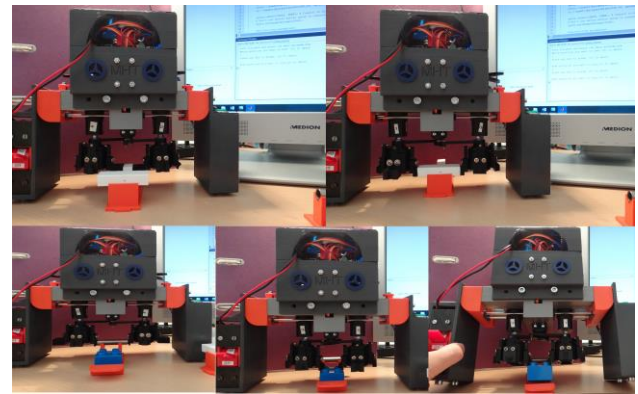


Bild 20 Prototyp des Mehrteilgreifers mit gegriffenen Objekten

Dieser Mehrteilgreifer kann noch weiter optimiert und das Gewicht auf maximal 500 g reduziert werden. Wie Bild 21 zeigt, kann er mit zwei Einweg-Freilaufaglern von nur einem Servomotor realisiert werden.

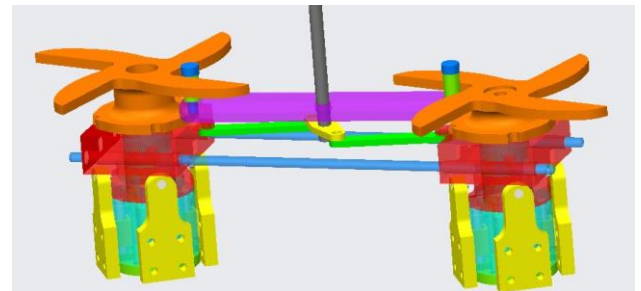


Bild 21 Optimierter Mehrteilgreifer

4 Zusammenfassung

Dieser Artikel enthält eine neuartige Symbologie und Definition für das Greifsystem, die eine einfache Erklärung für das komplexe Greifsystem ermöglicht. Im folgenden Kapitel wird ein Mehrteilgreifer (Minirevolver) für kollaborative Roboter entwickelt, optimiert und vorgestellt, der aus der Symbologie abgeleitet wird. In Zukunft wird die integrierte Einheit aus Level IIIIII entwickelt und getestet. Danach wird das dreistufige Greifsystem I-I/III-III erforscht, um die Produktivität des Pick & Place Prozesses zu steigern.

5 Literaturverzeichnis

- [1] VDI-Richtlinie 2740, Blatt 1: *Greifer für Handhabungsgeräte und Industrieroboter in Mechanische Einrichtungen in der Automatisierungstechnik*, Berlin, Beuth Verlag, 1995, S. 1-8
- [2] Hesse, S.: *Greifertechnik-Effektoren für Roboter und Automaten*, München: Carl Hanser, S. 149-152, 2011
- [3] Yin, F.: *Entwurf eines Mehrteilgreifers für ein neues Greifsystem*, Technische Universität Chemnitz, Bachelorarbeit, 2019

¹ Konstruktive Ausführung siehe [3]

- [4] Yesmukhametov, A.;Kalimoldayev, M.; Mamyrbayev, O. and Amirgeliev, Y.: *Design and Kinematics of Serial/Parallel Hybrid Robot*, in 2017 3rd International Conference on Control, automation and Robotics, 2017
- [5] Zeng,Q.;Fang,Y.: *Structural synthesis and analysis of serial-parallel hybrid mechanisms with spatial multi-loop kinematic chains*, Mechanism and Machine Theory, S. 198-215, 2011
- [6] Harib, K.H.;Moustafa,K.A.;Sharif Ullah,A. and Zenieh,S.: *Parallel, Serial and Hybrid Machine Tools and Robotics Structures:Comparative Study on Optimum Kinematic Designs*, in Serial and Parallel Robot Manipulators-Kinematics, Dynamics, Control and Optimization, Dr. Serdar Kucuk (Ed.), Croatia, InTech, ISBN: 978-953-51-0437-7, S. 109-122,2012
- [7] Monkman, G.J.;Hesse, S.;Steinmann, R. and Schunk, H.: *Robot Grippers*, Weinheim: WILEY-VCH, S.75-148,2007
- [8] Hesse, S.;Monkman, R.;Steinmann, R. and Schunk, H.: *Robotergriffe*, München Wien: Carl Hanser, S.80-161, 2004
- [9] Volmer, J.: *Getriebetechnik*, Berlin: VEB Verlag Technik, 1987

Reconfiguration of Modular Robots with triangular structure

Rekonfiguration modularer Roboter mit Dreiecksstruktur

Michael Gerbl, Leopold-Franzens-Universität Innsbruck, Institut für Mechatronik, 6020 Innsbruck, Österreich, michael.gerbl@uibk.ac.at

Michael Pieber, Leopold-Franzens-Universität Innsbruck, Institut für Mechatronik, 6020 Innsbruck, Österreich, michael.pieber@uibk.ac.at

Johannes Gerstmayr, Leopold-Franzens-Universität Innsbruck, Institut für Mechatronik, 6020 Innsbruck, Österreich, johannes.gerstmayr@uibk.ac.at

Kurzfassung

In diese Arbeit stellen wir Strategien für die Rekonfiguration von modularen Robotern mit dreieckiger Struktur vor. Wir analysieren die Kinematik von PARTS, einem modularen Roboter, der aus adaptiven Dreieckszellen aufgebaut ist, und präsentieren ein kinematisches Modell basierend auf einer Kette von parallelen Bipod-Manipulatoren mit zwei Freiheitsgraden. Unter Verwendung von Metamodulen, die aus vier einzelnen, in einer quadratischen Geometrie angeordneten Modulen bestehen, entwickeln wir 3 Bewegungsprimitive und zeigen die Äquivalenz zu einem modularen Roboter mit schwenkenden quadratischen Zellen, wodurch die Anwendung von Rekonfigurationsalgorithmen dieser Klasse von modularen Robotern auf modulare Roboter mit Dreiecksstruktur ermöglicht wird. Darüber hinaus schlagen wir Strategien für die Anwendung auf unstrukturierte Netze vor und skizzieren die nächsten Forschungsschritte.

Abstract

In this paper, we present strategies for the reconfiguration of modular robots with triangular structure. We analyse the kinematics of PARTS, a modular robot, consisting of adaptive triangular cells and present a kinematic model based on a chain of two degree of freedom bipod parallel manipulators. Utilizing meta-modules consisting of four modules arranged in a square geometry, we develop three motion primitives and show the equivalence to a pivoting square modular robot, thus enabling the application of reconfiguration algorithms designed for this class of modular robot. Additionally, we propose strategies for the application to unstructured meshes and outline the next research steps.

1 Introduction

Modular robots are mechatronical systems, consisting of multiple homogeneous or heterogeneous robotic cells, which can be arranged in a variable topology. Special interest receive self-reconfigurable modular robots with homogeneous cells and a lattice-type architecture, as such systems can by means of miniaturization serve as a basis for the development of programmable matter [8, 9].

Several different research groups currently work on this specific class of modular robots, often referred to as metamorphic robotic systems [4, 12, 19]. The elementary robotic cells of these systems generally consist of a regular geometrical shape, which can be arranged in a lattice based, space filling manner e.g. square [15, 17], circular [9] or hexagonal [12] modules in planar modular robots (2D) and cubical [14, 19] or spherical [5, 21] modules in spacial modular robots (3D), respectively.

The challenge of self-reconfiguration is the autonomous transformation from one topological configuration to a different one by systematic movement of individual modules. As the type of cell locomotion depends on the system design and differs vastly from system to system, every robot needs an individual approach to realize cell locomotion and structural reconfiguration. Current approaches to realize self-reconfiguration typically utilize one or more of the following principles: (i) meta-modules [3, 5, 15, 19], (ii) lo-

comotion along boundary surfaces [3, 5], (iii) movement of unoccupied voids ("holes") in the interior of the robot structure [6, 9] and (iv) scaffolding [7, 16]. Additionally, all modular robots rely on motion primitives for (meta-) module locomotion, as the motion capabilities of a single robotic cell are typically strongly limited. However, the topological structure of modular robots with a lattice-type architecture is due to the module design highly regular and motion primitives can rely on a predefined boundary to the remaining robot structure, reduced by symmetries to a discrete number of different configurations to consider.

The planar metamorphic robotic system *PARTS* [13], currently under development at the University of Innsbruck, is composed of adaptive triangular cells (ATCs), which can change the length of each edge by linear actuation. The modules can establish connections along edges and are therefore able to reproduce every planar triangular mesh within the geometrical limitations of the modules, regarding edge lengths and interior angles. Since triangular meshes are generally unstructured, autonomous self-reconfiguration can not rely on the structural regularity of a lattice and hence must cope with challenges of a priori unknown local environment including skewed triangular cells or geometrically locked modules, which can not contribute to the locomotion process of other cells.

In this paper we present strategies for the autonomous self-reconfiguration of *PARTS*. We first analyse the kinematics

and geometrical limitations of one ATC and establish minimal requirements for single cell locomotion. Building upon the locomotion criteria, we introduce a method for visualizing and analysing the topological structure of arbitrary valid triangular meshes using graph theory. In order to find solution strategies for the general reconfiguration problem of unstructured meshes, we first demonstrate that the reconfiguration for a structured mesh with square-shaped meta modules is in the absence of certain inadmissible subconfigurations possible. Subsequently we propose solution strategies with applicability for unstructured meshes in give an outlook for future research.



Figure 1 PARTS system consisting of six connected ATCs.

2 Kinematics of PARTS

2.1 Mechanical desing of PARTS

The **Planar Adaptive Robot with Triangular Structure (PARTS)** is a modular robot with self-reconfiguration capability, designed and constructed at the University of Innsbruck (LFUI). *PARTS* consists of many homogeneous **Adaptive Triangular Cells (ATC)** with three degrees of freedom, which can change their shape by altering the edge lengths via linear actuation. Each edge is equipped with a connection mechanism, that enables two ATCs to establish a connection along one edge. Due to the design of the connection mechanism, both edges must have the same length, which we will refer to as *connectivity constraint*. The revolute joint at the vertices of the modules is realised by a thoroughly designed six-bar mechanism, which shifts the center of rotation outside the extend of the mechanical structure. Hence, multiple connected ATCs can share the same vertex, thus enabling the construction of arbitrary planar triangular meshes within the geometrical limitations of the modules. The *PARTS* system is exemplary depict in **figure 1** with a system composed of six connected ATCs. The current ATC design allows an effective elongation of the idealized triangular edges of 43.4 %, corresponding to a minimum interior angle of 44.2° and a maximum interior angle of 91.6°. For further information and details regarding the mechanical design of ATCs and *PARTS*, see [13].

2.2 Kinematics of one ATC

PARTS can be represented by a kinematic tree with interconnected kinematic chains consisting of multiple ATCs, schematically depicted in **figure 2**. It is therefore convenient to describe the kinematics of a single ATC as a planar **RPRPR** (bipod) parallel manipulator, with two degrees of freedom, as shown in **figure 3**, since the length of the base edge c is due to the connectivity constraint determined by the preceding module in the kinematic chain. (Note: without loss of generality, the base edge of the ATC, which constitutes the root of the kinematic tree, can be regarded as constraint in length, position and orientation, such that the system is fully determined.) Thus, each ATC adds two degrees of freedom to the *PARTS* system, which allow to change the position of the unconstrained vertex C in the xy -plane (see figure 3).

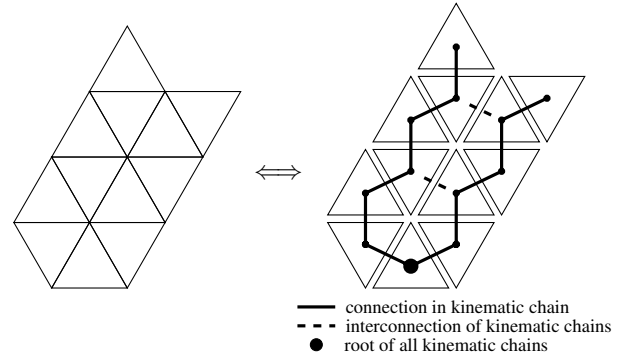


Figure 2 Representation of *PARTS* as kinematic tree with interconnected kinematic chains.

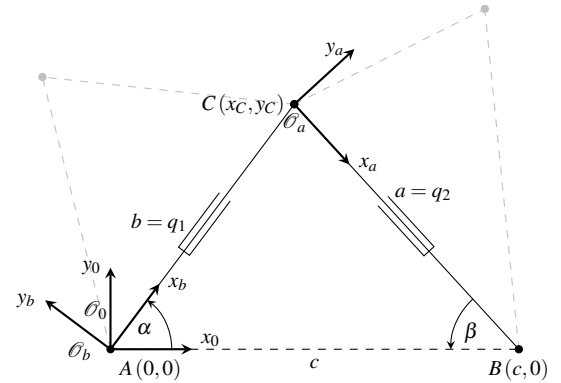


Figure 3 Kinematics of one ATC and local coordinate systems $\mathcal{O}_a$ and $\mathcal{O}_b$. Connected ATCs are schematically depicted by gray dotted lines.

A planar **RPRPR** parallel manipulator is composed of two connected **RPR** serial manipulators, fixed at point A and B , respectively. The coordinates of the vertex $C(x_C, y_C)$ in the local ATC coordinate system $\mathcal{O}_0$ with origin at point A and positive x -axis along the constraint edge c is given by

$$x_C = \frac{q_1^2 + c^2 - q_2^2}{2c}, \quad y_C = \sqrt{q_1^2 - x_C^2}, \quad (1)$$

with the length of the constraint base edge c and the driven linear actuator lengths q_1 and q_2 [18]. We can assign a

coordinate system $\mathcal{O}_a$ and $\mathcal{O}_b$ to the unconstrained edges a and b , respectively, with the x -axis pointing along the edge and the y -axis pointing away from the ATC. The transformation from frame $\mathcal{O}_0$ to $\mathcal{O}_a$ and $\mathcal{O}_b$ is described by the homogeneous transformation matrix T_a and T_b , respectively, defined as

$$T_a = \begin{pmatrix} \cos(-\beta) & -\sin(-\beta) & x_c \\ \sin(-\beta) & \cos(-\beta) & y_c \\ 0 & 0 & 1 \end{pmatrix}, \quad (2)$$

$$T_b = \begin{pmatrix} \cos(\alpha) & -\sin(\alpha) & 0 \\ \sin(\alpha) & \cos(\alpha) & 0 \\ 0 & 0 & 1 \end{pmatrix}. \quad (3)$$

The angles α and β can be expressed in terms of the positions vectors $\vec{b} = \vec{AC} = (x_c \ y_c)^T$ and $\vec{c} = \vec{AB} = (c \ 0)^T$ and $\vec{CB} = \vec{c} - \vec{b}$ by

$$\alpha = \text{atan2}(AC_y, AC_x) = \text{atan2}(y_c, x_c), \quad (4)$$

$$\beta = -\text{atan2}(CB_y, CB_x) = -\text{atan2}(-y_c, (c - x_c)), \quad (5)$$

where AC_x , AC_y , CB_x and CB_y are the x and y coordinates of vector $\vec{AC}$ and $\vec{CB}$, respectively.

2.3 Kinematics of connected ATCs

Given a kinematic tree representing the topological structure of *PARTS* in a distinct configuration, the kinematics of *PARTS* can be fully described by a sequence of local transformations (2) and (3). For example, considering a kinematic chain with a length of $i + 1$ ATCs, with the root module having the index 0 and the $i + 1$ module in the chain the index i , the transformation ${}^{0i}T$ from the frame of the root ATC ${}^0\mathcal{O}_0$ to the frame ${}^i\mathcal{O}_0$ of ATC i can be calculated by

$${}^{0i}T = \prod_{k=0}^{i-1} {}^kT_s, \quad s = \{a, b\}, \quad (6)$$

where kT_s is the local transformation for triangle k in the considered kinematic chain from ${}^k\mathcal{O}_0$ to ${}^k\mathcal{O}_a$ or ${}^k\mathcal{O}_b$, respectively, depending on the local edge triangle $k + 1$ is connected to. The coordinates x_i and y_i of the unconstrained vertex C_i of triangle i is thus given by

$$x_i = ({}^{0i}T \cdot {}^iT_a)_{13} \quad \text{and} \quad y_i = ({}^{0i}T \cdot {}^iT_a)_{23}. \quad (7)$$

The interconnection of kinematic chains imposes constraints on the edge lengths of the ATCs involved in the connection. This is illustrated in **figure 4**, which depicts the lower part of figure 2. Due to the interconnection between the two kinematic chains at ATC 5 and 6 the position of vertices 5C and 6C of ATC 5 and 6, respectively, must coincide. Thus, one ATC is completely constrained and can not contribute to the overall degrees of freedom of the *PARTS* system.

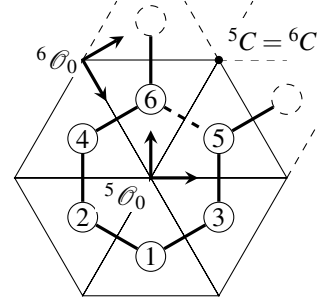


Figure 4 Interconnected kinematic chains with a connection between ATC 5 and 6.

2.4 Locomotion of ATCs

The basic method of locomotion for a single ATC consists in forming a new connection to a nearby ATC and then terminating all connections besides the newly formed one. In order to establish a new connection between two ATCs m and n along the local edges e_m and e_n , which will be referred to as *connecting edges*, the coordinates of the vertices of both connecting edges must coincide, schematically depicted in **figure 5**. However, since the positioning accuracy of the current ATC design is limited and strongly decreases with an increasing number of modules in a kinematic chain due to error accumulation [13], only ATCs which initially have one vertex in common are considered to form new connections. (Note: the connection of two kinematic chains by the previously described method is nevertheless necessary to form topological structures with enclosed free surfaces; this, however, is not in the scope of this paper.)

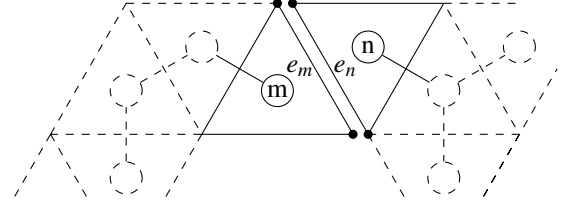


Figure 5 Connection between two ATC m and n , which belong to different kinematic chains and have no vertex in common.

The connection procedure for two ATCs with one vertex initially coinciding can be regarded as forming a *cycle* or closed kinematic chain around that vertex, as shown in **figure 6**. In order to form a cycle around a given vertex, there must be a minimum number of four ATCs around that vertex, since the maximum interior angle of an ATC is limited to an angle just slightly above 90° (see section 2.1). This necessary condition to form a new connection will be referred to as *closing condition*. Thus, the locomotion process can be summarized by two sequential steps, which are illustrated in figure 6 exemplary with four ATCs, where ATC 1 is moved and ATC 2, 3 and 4 are referred to as *stationary* regarding the locomotion of ATC 1, since their positions before and after the movement are identical:

1. *Closing*: Change actuator lengths in order to form a cycle and connect triangles along connecting edge
(a) $\Rightarrow$ (b), connecting ATC 1 and 4

2. *Opening*: Terminate prior connection(s) and restore actuator lengths of stationary ATCs

(b) $\Rightarrow$ (c), disconnecting ATC 1 and 2

As can be seen in figure 6, the closing procedure generally leads to shape alteration of all involved ATCs, emphasizing the closing condition not to be a sufficient condition for locomotion. Therefore the local environment of the involved ATCs must also be taken into account in order to ensure, that the forming of a cycle is not restrained by ATCs with reduced motion capability through connectivity constraints. Furthermore, in order to form new connections, ATCs need to have free edges without other ATCs attached, thus locomotion is only feasible along the boundary surface of *PARTS*.

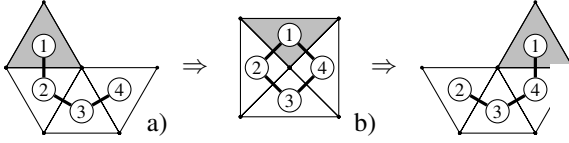


Figure 6 Locomotion of ATC 1 from ATC 2 (a) to ATC 4 forming a cycle 1-2-3-4 (b)

2.5 Topological representation of *PA* using graphs

Graph theory is a powerful tool to structure and analyze pairwise relation between objects. In the field of mechatronics graphs are often used for motion planning or connectivity analyses of robots with a chain-type structure [10]. In this paper, we will use graphs to visualize the topology of *PARTS* and set a framework for future research. Each ATC is represented by a *node* and a connection between two modules is a *edge* of the graph. The graph generated in this manner is a planar graph, since no edges intersect. A *cycle* is defined as a closed path through the edges of a graph, where chordless cycles (no two edges of the cycle are connected by an edge which does not belong to the cycle) with a length between 4 and 8 are of special interest, as such cycles are equivalent to the kinematic chains / cycles around an ATC vertex, defined in section 2.4. The *degree* of a node is given by the number of edges (connections) and ranges from 1 to 3. Color coding of nodes depending on the node degree highlights ATCs with free edges, which can form a new connection. A *spanning tree* of the graph is a subgraph containing all the nodes of the graph with a minimum possible number of edges. Thus, since *PARTS* can always be represented by multiple interconnected kinematic chains as we conducted in section 2.2, the spanning tree of the graph represents one possibility of such kinematic chains and the difference between the original graph and the spanning tree corresponds to all interconnection between the individual chains, thus yielding an efficient way to generate the kinematic model of *PARTS*. A triangular mesh and the corresponding graph representing the topology is depicted in figure 7. The spanning tree is drawn by blue lines, while the dashed blue lines represent the additional connections. Nodes with degree less than 3

(red and yellow) can form potential cycles, as long as they share a vertex and satisfy the closing condition (new cycle contains at least four nodes). However, the representation as a graph is not unique, as open kinematic chains with more than one element contain ambiguities, since the location of the free triangle edge of a node with degree 2 (two connections) with respect to the kinematic chain is not known from the graph alone (e.g. in figure 7 in the kinematic chain 7 – 8, the location of the free edge of ATC 7 could be on the right or left of the kinematic chain, which is from the graph alone not apparent). One possibility to resolve this ambiguity is to use connector graphs similar to [10], wherever the precise connectivity is not vivid. As we always plot the corresponding modules in addition to the graph, we omit the use of connector graphs for sake of figure readability.

In addition, graphs can be used to reveal the similarities of different configurations by finding the maximum common

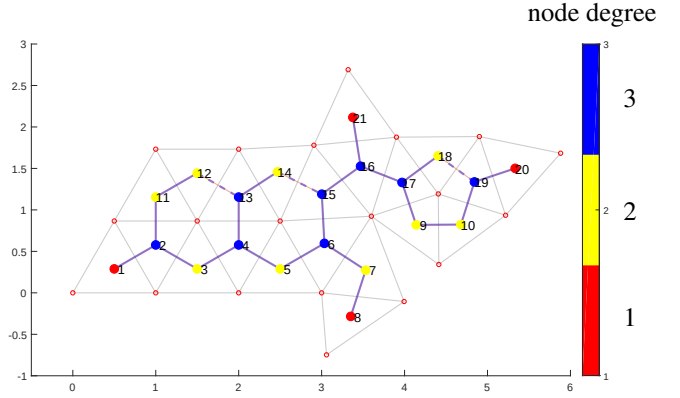


Figure 7 Representation of the topology of *PARTS* using graphs

3 Reconfiguration strategies

A frequently applied method for structural reconfiguration consists in moving modules along the boundary surface of the robot to specified goal positions [3, 5]. Alternative reconfiguration approaches achieve a change in topology by the movement of unoccupied voids through the interior of the robot structure [6, 9] or the use of a definite scaffold [7, 16] for module arrangement, in order to leave enough clearance for individual modules to traverse through the structure. In this paper, we will focus on the locomotion of ATCs along the boundary surface of the robot, as the other methods necessarily require the formation of topological structures with enclosed free surfaces, which was evaluated as unpractical in section 2.4 for the current ATC design.

As no reconfiguration algorithm for robots with triangular structure exists to the authors knowledge in the present literature, we will derive in this section a reconfiguration algorithm for *PARTS* utilizing square meta-modules by development of motion primitives, which are equivalent to pivoting motions. In the end of this section, we will give an

outlook for strategies, which may be useful to achieve the reconfiguration of unstructured meshes.

3.1 Reconfiguration of rectangular meshes utilizing meta-modules

As the motion capabilities of a single module are typically strongly limited, many authors propose the usage meta-modules [3, 5, 15, 19]. A meta-module is formed by grouping together multiple modules to an autonomous entity with increased motion capability compared to a single module. **Figure 8** shows a possible meta-module shape for *PARTS*, consisting of four ATCs arranged in a square geometry. Such meta-modules can be organized in a orthogonal lattice, as schematically depicted in the figure 8. Two connected meta-modules always satisfy the closing condition, since each meta-module corner is formed by two connected ATCs. However, as the individual ATCs are close to their actuation limits (interior angles either 90° or 45°), one meta-module in its closed form, that is the four ATCs are connected and form a cycle around the vertex, is practically stiff and can not contribute to motion process of other modules. In the next sub-section we will present three motion primitives (MP), sufficient for reconfiguration within a subset of a configuration. The motion primitives will further elaborate a possible way to deal with constrained ATCs during the locomotion process.

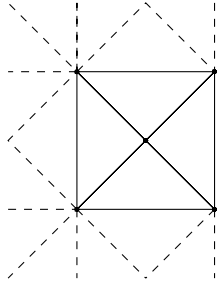


Figure 8 Meta-module consisting of four ATCs arranged in square geometry, schematically embedded in a grid of modules

3.1.1 MP1: advancing on a straight edge

The first motion primitive consists in moving along a straight edge, as illustrated in **figure 9**. Only the solid drawn meta-modules are actively involved in the motion process, where meta-module 5-6-7-8 is moved one position downwards with the assistance of meta-module 1-2-3-4. In order to advance one position, the meta-modules have to first generate mobility by terminating one connection within the meta-module each. In the figure, ATCs 1-2 and 5-6, respectively, detach from one another (a). Then, ATC 1 and 6 can move their edges together and establish a connection (b). After detachment of meta-module 5-6-7-8 from the remaining structure by terminating all connections of ATC 7 and 8 to other meta-modules, ATCs 1-2 and 5-6 can connect again and restore the original meta-module shape (c). Note, that this movement is equivalent to pivoting me-

ta-module 5-6-7-8 around one of its corners, requiring additional clearance adjacent to the initial and final position, marked as gray area in figure 9 (b).

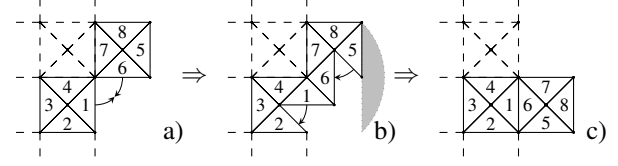


Figure 9 MP1: advancing on a straight edge. Only the solid modules are involved in the motion, the shaded area is swept while moving meta module 5-6-7-8

3.1.2 MP2: pivoting around corner (one connection)

The second motion primitive consists in pivoting a meta-module around the corner of a stationary meta-module,

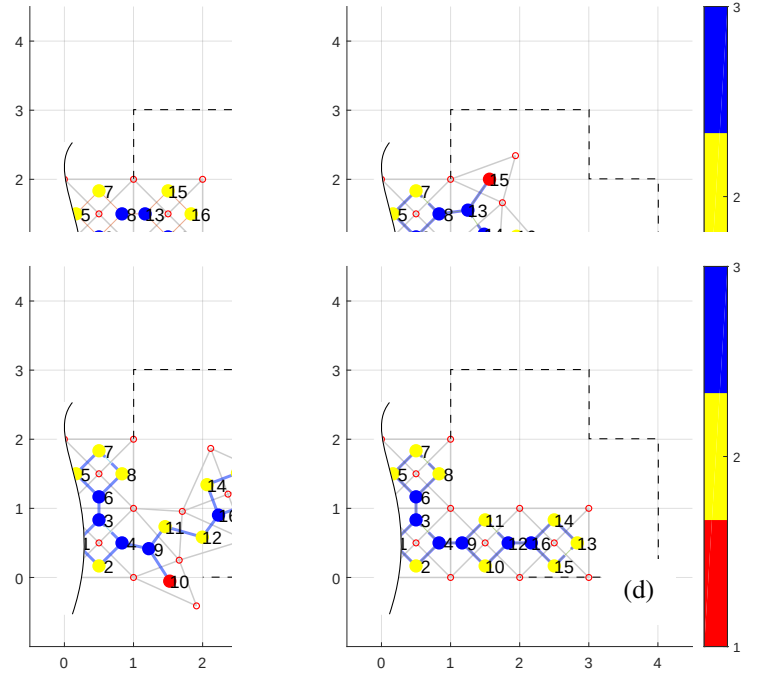


Figure 10 MP2: pivoting around a corner, where (a) is transformed by subsequent detachment /attachment of ATCs through (b) and (c) to (d); meta-module 13-14-15-16 is moved with assistance of the meta-module 9-10-11-12. The dotted line represents the boundary to adjacent lattice positions, which need to be unoccupied during the motion process

Meta-module 13-14-15-16 is pivoted with assistance of meta-module 9-10-11-12 around the corner to the lattice position right of meta-module 9-10-11-12. Similar to MP1, we first have to generate mobility by terminating the intra-meta-module connections between ATC 10-12 and 15-16, respectively, so that ATC 12 and 16 can establish a new connection (b). Notice, that there needs to be space on the bottom of meta-module 9-10-11-12, as ATC 10 sweeps through the adjacent lattice position due to deformations of the other ATCs. Next, the ATC connections 8-13 and 11-14 are terminated so meta-module 13-14-15-16 can reattach ATC 15 and 16 to restore the original meta-module shape

(c). In the last step, the stationary meta-module 9-reattaches ATC 10 and 12 to restore the meta-module finish the movement. The dashed line in figure 11 represents the border to adjacent lattice positions, which must be unoccupied during the motion process.

3.1.3 MP3: pivoting around corner (two constraints)

The third motion primitive also pivots a meta-module around the corner of another meta-module and is similar to MP2. However, the stationary meta-module is constrained on two edges by connections to the robot, compared to only one for MP2. The motion for MP3 is depicted in **figure 12**. Meta-module 21-17-18-19-20, in order to move to the free position to the right of the stationary meta-module (c), the stationary meta-module is constrained on two edges. First, we need to generate space for ATC 20 and 24 to move and form a new cycle. Hence, we terminate the connection between ATCs 12-21, 8-17 and 13-15, respectively. We use ATC 15 to tip the meta-modules 17-18-19-20-22-23-24 to the right, thus creating space at the corner. In the next step, we disconnect ATCs 18-20 and 17-22, respectively, and establish a connection between ATC 17-24 in a similar manner as for MP2 (c). Detaching from 19 (d) allows to restore meta-module 21-22-23-24 to its original shape (e). In the subsequent steps ATC 17 is brought to its original shape (f), to avoid collision when ATC 13 and 15 are reconnected. The last step consists in ATCs 18 and 20 connecting to each other and thus completing the motion by restoring the shape of meta-module 17-18-19-20 (h).

The dotted lines in figure 12 show the boundaries of adjacent lattice positions, which need to be unoccupied for MP3 to be successfully performed. This space, however, is similar for a pivoting module motion, as depicted in **figure 11**. Thus, MP3 (and also MP2, since the same unoccupied space is required) are equivalent to a simple pivoting of one square around the corner of another square with regards to clearance requirements.

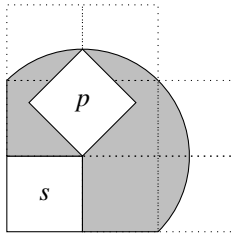


Figure 11 Swept area by pivot movement of a square (p) around the corner of another stationary square (s)

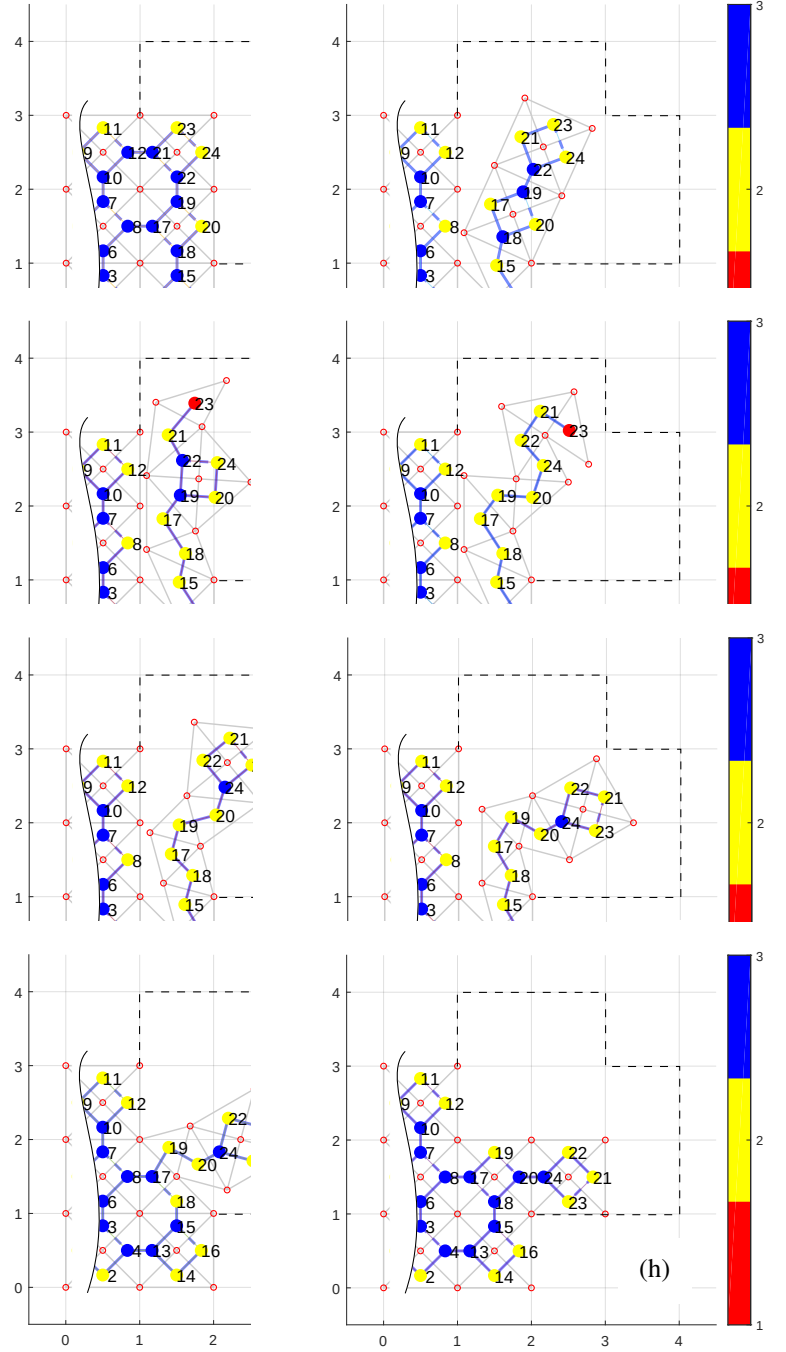


Figure 12 MP3: pivoting around a corner, where (a) is transformed by subsequent (de-)attachment of ATCs through (b-g) to (h); meta-module 21-22-23-24 is moved with assistance of the meta-module 17-18-19-20. The dotted line represents the boundary to adjacent lattice positions, which need to be unoccupied during the motion

3.1.4 Complete reconfiguration for rectangular mesh

Regarding necessary unoccupied lattice positions, which are needed as clearance during the motion process, the motion primitives MP1-MP3 are equivalent to a pivoting motion of one square around one of its corners shared with another square. Thus, reconfiguration algorithms for pivoting squares with strong connectivity, meaning connectivity through edges, can be directly implemented for the reconfiguration of the *PARTS* system with square meta-modules. One such complete algorithm is given by [17],

guaranteeing reconfiguration of an admissible subset of all configurations in $O(n^2)$ pivoting moves, where n is the number of square meta-modules.

Inadmissible subconfigurations are given in **figure 13**. If none of these subconfigurations (up to rotation and mirroring) is present in a given configuration, the reconfiguration is always possible.

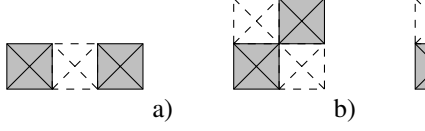


Figure 13 Inadmissible subconfiguration, where positions are depict by gray meta-module positions by outlined meta-modules

3.2 Strategies for unstructured

In the following section, we will deduce configuration strategies for unstructured results of the previous section. In addition a method for cell locomotion, even when condition is not satisfied.

3.2.1 Selective disconnection

As we saw in section 3.1.3, motion constraints can be overcome by *selective* modules. Constrained ATCs can be sequenced from the robotic system, doing this in a kinematic subchain, which is increasing each disconnection is emerging. The disconnection progress up to a so called *pivoting module* fully constrained and can act as center of a newly formed kinematic subchain, being a complete kinematic chain and generate reconfiguration, as was already demonstrated in step (a) to (b).

3.2.2 Cycle integration and inverse cycle

In this subsection, we will refer to *cycle* integrating one connection in a cycle and connecting in such a way, that space is generated between connected edges, and *cycle closing* as access in bringing potential connecting edges vertex together and thus forming a new cycle. A thorough combination of the operation and cycle closing enables integration of the surface into an existing cycle (*cycle integration*). Cycle integration is demonstrated by integrating ATC 8 from the surface into a cycle. From a starting configuration in (a) a cycle opening is performed at the connection 7-9, opening the cycle 7-6-15-16-17-7-9 (b). Another cycle opening of 17-18-19-10-9 by disconnecting 17-9 (c) creates space to perform a cycle closing of cycle 7-6-25-26-27 by connecting ATC 7 and 17 (d). The

next step consists in cycle closing of 8-7-17-18-19-10-9 by connecting ATC 8 and 9 (e) and subsequently opening the same cycle by disconnection of ATC 7 and 8 (f). Another cycle opening of cycle 7-6-15-16-17 by detaching ATC 7 and 17 (g) and subsequent cycle closing of 8-9-10-19-18-17 and 7-6-5-16-17-18-8 yields (h) and the desired result to integrate ATC 8 into the cycle 7-9-10-19-18. To perform

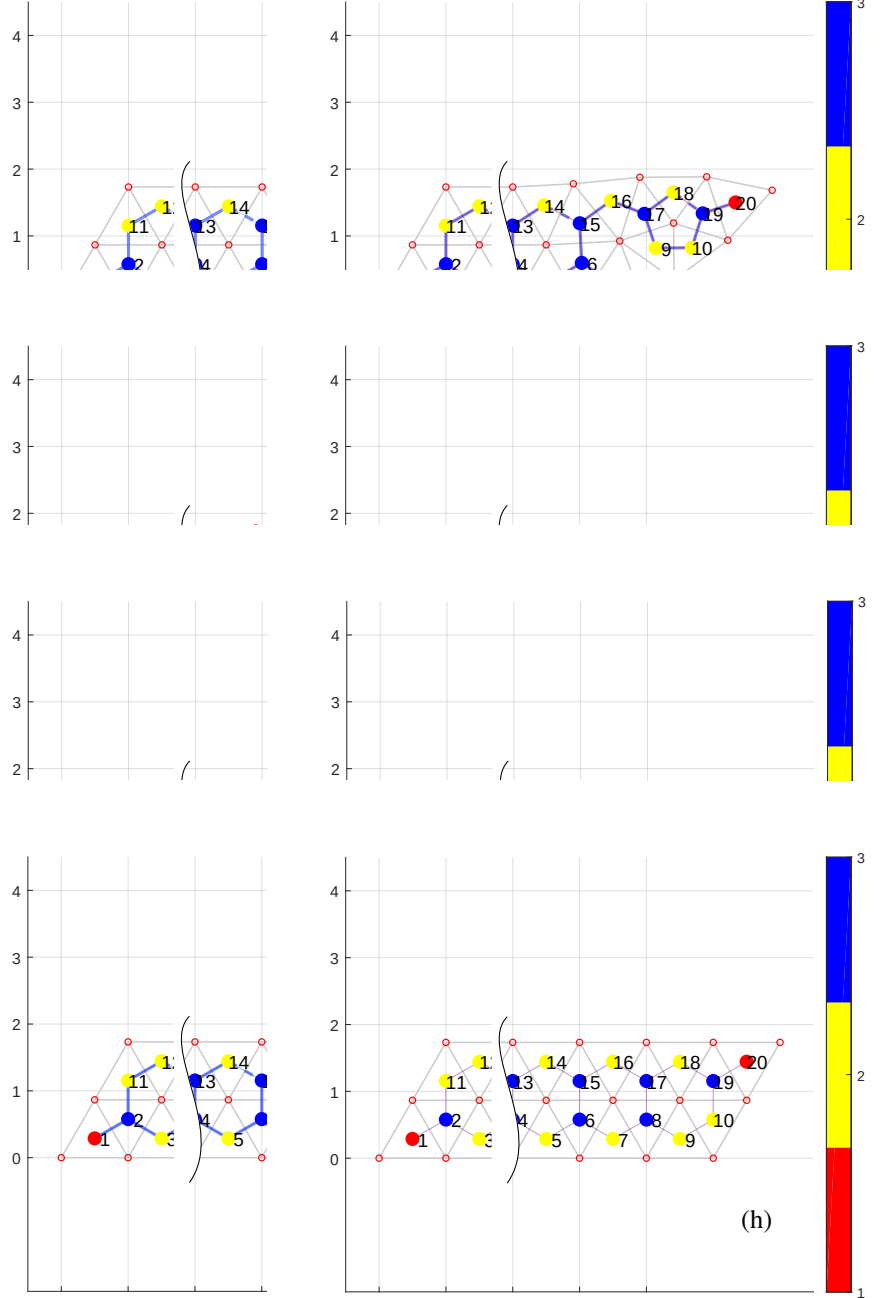


Figure 14 Cycle integration of ATC 8 performed by subsequent cycle opening/closing operations

4 Summary

In this paper, we demonstrated a complete reconfiguration algorithm for modular robots with triangular structure, designated for *PARTS*, using meta-modules, consisting of

4 ATCs. We developed 3 motion primitives and showed their equivalence to pivoting square motion, thus allowing the application of reconfiguration algorithms for pivoting square modular robots [17].

We also sketched strategies for reconfiguration, which are applicable for unstructured *PARTS* meshes. In future research, we will elaborate the proposed strategies with the goal of a general reconfiguration approach for modular robots with triangular structure, which can be performed on unstructured meshes.

5 Literatur

- [1] Asadpour, M.; Sproewitz, A.; Billard, A.; Dillenbourg, P.; Ijspeert, A.: *Graph Signature for Self-Reconfiguration Planning* IEEE/RSJ International Conference on Intelligent Robots and Systems, 2008, pp. 863–869
- [2] Chiang, C.; Chirikjian, G.: *Modular Robot Motion Planning Using Similarity Metrics* Autonomous Robots. (2001) 10-1, pp. 91–106
- [3] Chirikjian, G.: *Kinematics of a metamorphic robotic system*. Proceedings of the 1994 IEEE International Conference on Robotics and Automation, pp. 449–455
- [4] Chirikjian, G.; Stein, D.; Pamecha, A.; Chiang, C.: *Design and Implementation of metamorphic robots*. Proceedings of The ASME Design Engineering Technical Conference and Computers in Engineering Conference, October 1996, pp. 1–10
- [5] Christensen, D. J.; Stoy, K.: *Selecting a meta-module to shape-change the ATRON self-reconfigurable robot*. Proceedings - IEEE International Conference on Robotics and Automation, June 2006, pp. 2532–2538
- [6] De Rosa M.; Goldstein S.; Lee P.; Campbell J.; Pillai P.: *Scalable shape sculpting via hole motion: Motion planning in lattice-constrained modular robots* Proceedings of the 2006 IEEE International Conference on Robotics and Automation, , pp. 1462-1468
- [7] Dewey, D.; Ashley-Rollmann, M.; De Rosa, M.; Goldstein, S.; Mowry, T.; Srinivasa, S.; Pillai, P.; Campbell, J.: *Generalizing Metamodules to Simplify Planning in Modular Robotic Systems* IEEE/RSJ International Conference on Intelligent Robots and Systems, 2008, pp. 1338–1345
- [8] Gilpin, K.; Knaian, A.; Rus, D.: *Robot Pebbles: One Centimeter Modules for Programmable Matter through Self-Disassembly*. Proceedings of the 2010 IEEE International Conference on Robotics and Automation, pp. 2485–2492
- [9] Goldstein, S.; Campbell, J.; Mowry, T.: *Programmable Matter*. IEEE Computer. (2005) 38-6, pp. 99–101
- [10] Hou, F.; Shen, W.: *Graph-based optimal reconfiguration planning for self-reconfigurable robots*. Robotics and Autonomous Systems (2014) 62-7, ISSN 0921-8890, pp.1047–1059
- [11] Murata, S.; Yoshida, E.; and Kamimura, A.; Kurokawa, H.; Tomita, K.; Kokaji, S.: *M-TRAN: Self-reconfigurable modular robotic system*: IEEE/ASME Transactions on Mechatronics (2002), 7-4, ISSN 10834435, pp. 431–441
- [12] Pamecha, A.; Chiang, C.; Stein, D.; Chirikjian, G.: *Design and Implementation of metamorphic robots* Proceedings of The ASME Design Engineering Technical Conference and Computers in Engineering Conference, August 1996, Irvine, pp. 1–10
- [13] Pieber, M.; Neurauter, R.; Gerstmayr, J.: *An adaptive robot for building in-plane programmable structures* IEEE/RSJ International Conference on Intelligent Robots and Systems 2018, October 2018, Madrid, pp. 5321–5327
- [14] Romanishin, J. W.; Gilpin, K.; Rus, D.: *M-blocks: Momentum-driven, magnetic modular robots*: 2013 IEEE/RSJ International Conference on Intelligent Robots and Systems, Tokyo, 2013, pp. 4288–4295.
- [15] Rus, D.; Vona, M.: *Crystalline Robots: Self-Reconfiguration with Compressible Unit Modules*. Autonomous Robots (2001) 10-1, ISSN 1573-7527, pp. 107–124
- [16] Stoy, K.; Nagpal, R.: *Self-Repair Through Scale Independent Self-Reconfiguration* IEEE/RSJ International Conference on Intelligent Robots and Systems, 2004, pp. 2062–2067
- [17] Sung, C.; Bern, J.; Rosmanishin, J.; Rus, D.: *Reconfiguration Planning for Pivoting Cube Modular Robots* IEEE Int. Conference on Robotics and Automation 2015, May 2015 Seattle, pp. 1933–1940
- [18] Szept, C.; Stan, S.; Csibi, V.; Manic, M.; Balan, R.: *Kinematics, Workspace, Design and Accuracy Analysis of RPRPR Medical Parallel Robot*. Proceedings of the 2nd Conference on Human System Interactions, HSI, June 2009, pp. 75–80
- [19] Vassilvitskii, S.; Yim, M.; Suh, J. W.: *A Complete, Local and Parallel Reconfiguration Algorithm for Cube Style Modular Robots*. IEEE Int. Conference on Robotics and Automation 2002, May 2002 Washington DC, pp. 117–122
- [20] Vassilvitskii, S.; Kubica, J.; Rieffel, E.: *On the General Reconfiguration Problem for Expanding Cube Style Modular Robots Serguei*. IEEE Int. Conference on Robotics and Automation 2002, May 2002 Washington DC, pp. 801–808
- [21] Meng, Y.; Zhang, Y.; Sampath, A.; Jin, Y.; Sendhoff, B.: *Cross-Ball: A New Morphogenetic Self-Reconfigurable Modular Robot* . IEEE Int. Conference on Robotics and Automation, May 2011 Shanghai, pp. 267–272

Pre-Processing für taskbasierte Kinematik-Synthese

Pre-Processing for Task-Based Kinematic Synthesis

Veronika Zumpe, KUKA Deutschland GmbH, 86165 Augsburg, Deutschland, veronika.zumpe@kuka.com

Johannes Horeis, Rheinisch-Westfälische Technische Hochschule Aachen, Lehrstuhl und Institut für Getriebetechnik, Maschinendynamik und Robotik, 52074 Aachen, Deutschland, johannes.horeis@rwth-aachen.de

Prof. Dr.-Ing. Mathias Hüsing, Rheinisch-Westfälische Technische Hochschule Aachen, Lehrstuhl und Institut für Getriebetechnik, Maschinendynamik und Robotik, 52074 Aachen, Deutschland, huesing@igmr.rwth-aachen.de

Univ.-Prof. Dr.-Ing. Dr. h. c. Burkhard Corves, Rheinisch-Westfälische Technische Hochschule Aachen, Lehrstuhl und Institut für Getriebetechnik, Maschinendynamik und Robotik, 52074 Aachen, Deutschland, corves@igmr.rwth-aachen.de

Kurzfassung

Für manche Handhabungsaufgaben im Automatisierungsbereich sind Sonder-Roboterkinematiken gefordert. Daher besteht die Notwendigkeit, ebendiese für spezielle Bewegungsaufgaben effizient zu synthetisieren. In diesem Beitrag wird der erste Teil einer Methode vorgestellt, mit der eine taskbasierte Kinematik-Synthese für Roboter-Regionalstrukturen teilautomatisiert erfolgen kann. Innerhalb des *Pre-Processings* wird die Aufgabenstellung durch Vorgabe von Arbeitsbereichen auf Basis eines Umgebungsmodells und optionalen Randbedingungen definiert. Anhand dieser Eingabedaten werden während der Verarbeitung Übergangsbereiche ermittelt, die frei von Kollisionen mit der Umgebung sind. Anschließend werden Punktmengen in den Arbeits- und Übergangsbereichen generiert und als Output-Daten klassifiziert ausgegeben. Diese Punkte können wiederum im nächsten Syntheseschritt für die Struktur- und Maßsynthese als Vorgabepositionen herangezogen werden. Die praktische Umsetzung der Methode erfolgt in dem Tool *PreSynth*, was einen *Pre-Processor* für eine Kinematik-Synthese darstellt.

Abstract

In case of automatization of specific handling-tasks, it is necessary to develop novel robots with special kinematics. Thus, there is a need to efficiently synthesize these robots for the required motion tasks. In this article, the first part of a method for a partially automated kinematic synthesis for regional structures of robots is presented. The task is defined by specifying workspaces based on an environment model and optional constraints. These input data are used to determine transition areas that are free from collisions with the environment. Based on this input data, point sets are generated in the work and transition volumes and will be classified as output data. These points can be used as input data for the structural and dimensional synthesis as default positions. The practical implementation of this method is done in the tool *PreSynth*, which represents the pre-processor for kinematic synthesis.

1 Einleitung

Damit eine Kinematik-Synthese effizient ablaufen kann, bietet sich ein rechnergestützter Syntheseprozess an. Dazu sind CAD-integrierte Softwarelösungen in der Entwicklung besonders gefragt [1]. Lässt sich der Auslegungsprozess außerdem in einen methodischen Entwicklungsprozess z.B. nach der VDI 2221 [2] einbinden, so kann der Entwickler systematisch durch die Kinematik-Synthese und Konstruktion geführt werden. Dadurch ist es möglich, die Entwicklungszeit von komplexen technischen Systemen zu reduzieren und den Konstrukteur bestmöglich zu unterstützen.

Für die Entwicklung einer neuen Kinematik werden verschiedene Arbeitsschritte durchlaufen (siehe **Bild 1**). Zu Beginn wird die Aufgabenstellung definiert. Diese bein-

haltet die Vorgabe der geforderten Bewegungsaufgabe sowie das Definieren der notwendigen Randbedingungen.

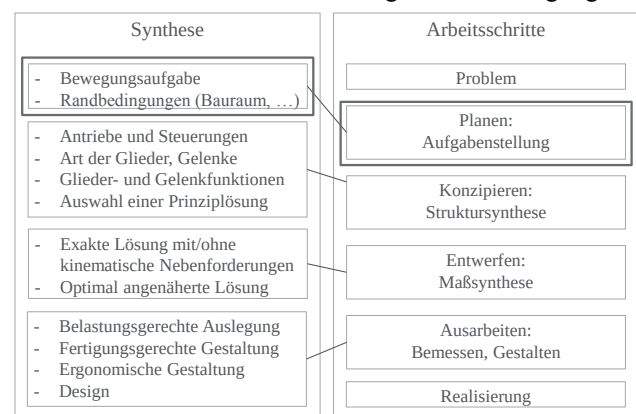


Bild 1 Arbeitsschritte einer Kinematik-Synthese

In Syntheseprogrammen übernimmt diese Arbeitsschritte oftmals der *Pre-Processor*. Anschließend folgen die Struktursynthese und darauf die Maßsynthese, welche von einem *Solver* übernommen werden können. Sind für die ausgewählte Prinzip-Lösung die Gliedlängen ermittelt worden, so erfolgt die Ausarbeitung und Konstruktion. Mit dem *Post-Processor* eines Syntheseprogramms wird das Ergebnis visualisiert und ausgewertet. [1,3]

Für das Realisieren spezieller Handhabungsaufgaben ist die Entwicklung von neuen Roboterkinematiken gefordert. Dies ist ein aufwändiger Entwicklungsprozess mit einer Vielzahl an Lösungsmöglichkeiten und erfordert in der Regel Erfahrungswerte und Fachwissen. Zwar gibt es Tools zur rechnergestützten Synthese oder Auswahl von Roboterkinematiken wie beispielsweise in [3-5] benannt, allerdings stehen nach Stand der Technik nur begrenzt Möglichkeiten zur computergestützten, taskbasierten Synthese für serielle Roboterkinematiken zu Verfügung. Daher wird in diesem Beitrag eine Methodik für eine taskbasierte Synthese für Roboter-Regionalstrukturen vorgestellt und beschrieben. Dabei liegt der Fokus auf dem ersten Arbeitsschritt, dem Definieren und Planen der Bewegungsaufgabe, was mittels *Pre-Processor* durchgeführt wird.

2 Methodik zur taskbasierten Synthese von Roboterkinematiken

Um für eine geforderte Bewegungsaufgabe eine geeignete Kinematik zu synthetisieren und zu gestalten, werden die Arbeitsschritte aus **Bild 1** systematisch in Verbindung gesetzt und teilautomatisiert, um den Anwender strukturiert und rechnergestützt durch den Entwicklungsprozess zu führen (siehe **Bild 2**).

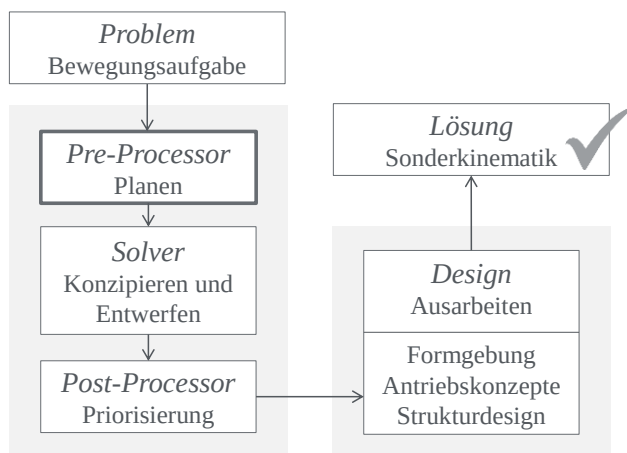


Bild 2 Methodik der taskbasierten Kinematik-Synthese

Durch die Vorgabe des Problems, welches durch die Bewegungsaufgabe beschrieben wird, kann mit Hilfe des *Pre-Processors* die zu erfüllende Aufgabenstellung nach interaktiver Vorgabe der Randbedingungen automatisiert geplant werden. Die daraus resultierenden Output-Daten werden als Vorgabepositionen einer Struktur- und Maßsynthese herangezogen, wodurch die Kinematik konzipiert und entworfen wird. Diese Arbeitsschritte werden in

dem *Solver* umgesetzt. Eine Auswahl von geeigneten Kinematiken wird im *Post-Processor* dargestellt. Nach Priorisierung erfolgt in der anschließenden Ausarbeitung das *Design* des Roboters. Dabei wird der Anwender im Strukturdisein und bei der Erstellung und Auswahl von Antriebskonzepten unterstützt. Als Output folgt die Grobkonstruktion der auf die Bewegungsaufgabe angepassten Sonderkinematik.

3 Lösungsstruktur des Pre-Processors

Der erste Teil der Methode zur Entwicklung von Roboter-Regionalstrukturen beinhaltet das *Pre-Processing* der Kinematik-Synthese und ist in **Bild 3** dargestellt.

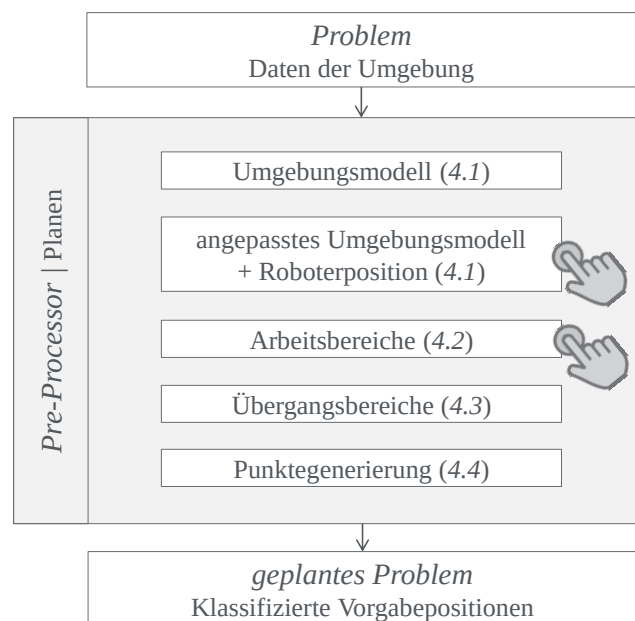


Bild 3 Lösungsstruktur des Pre-Processors

Als Input-Daten werden dem *Pre-Processor* Modelle aus einem CAD-Systemen übergeben, wodurch die Umgebung der Handhabungsaufgabe beschrieben wird. Das Umgebungsmodell kann optional innerhalb des *Pre-Processors* durch Hinzufügen weiterer Objekte und mittels Transformationen angepasst werden. Weiterhin erfolgt die Vorgabe des Aufstellortes der Roboterbasis (Roboterposition). Die eigentliche Handhabungsaufgabe wird durch das Definieren von Arbeitsbereichen und Arbeitspunkten beschrieben. Um sicherzustellen, dass die Arbeitsbereiche allesamt von einer später entwickelten Kinematik erreicht werden können, werden innerhalb des *Pre-Processors* im Verarbeitungsprozess paarweise Übergangsbereiche zwischen den einzelnen Arbeitsbereichen ermittelt. Die Übergangsbereiche sind frei von Kollisionen mit dem Umgebungsmodell und groß genug, sodass ein Objekt definierbarer Größe hindurchgeführt werden kann. Anschließend werden in den Arbeitsbereichen und Übergangsbereichen Punkte generiert, welche als Positionen für eine nachfolgende Struktur- und Maßsynthese innerhalb des *Solvers* herangezogen werden. Um den späteren Syntheseprozess effizient gestal-

ten zu können, werden außerdem besonders weit von der Roboterposition entfernte Punkte oder weit voneinander entfernte Punkte automatisch identifiziert. Als Output-Daten übergibt der *Pre-Processor* die generierten Punkte klassifiziert, sodass besonders aussagekräftige Punkte im weiteren Syntheseprozess priorisiert behandelt werden können, um das Lösungsfeld frühzeitig einzuschränken.

4 Lösungsprozesse des Pre-Processor

Die Methode zum *Pre-Processing* für eine Kinematik-Synthese wird durch das nachfolgend beschriebene Tool *PreSynth* in MATLAB umgesetzt (siehe **Bild 4**).

Für die Benutzereingaben verfügt das Tool über eine umfangreiche grafische Benutzeroberfläche, die es dem Anwender ermöglicht, interaktiv durch die Beschreibung und Planung der Handhabungsaufgabe geführt zu werden. Dazu können Umgebungsmodelle aus allen gängigen CAD-Programmen importiert, angezeigt und als Grundlage für weitere Eingaben und Operationen verwendet werden. Auf Basis dessen erfolgt die Verarbeitung und als Output liefert der *Pre-Processor* Vorgabepositionen, welche wiederum die Input-Daten des *Solvers* bilden.

Nachfolgend wird zuerst die verwendete Umgebungsrepräsentation erläutert und anschließend beschrieben, wie Arbeitsbereiche innerhalb dieser Modelle definiert werden. Nach den Nutzereingaben erfolgt die Verarbeitung und es werden die Übergangsbereiche ermittelt. Dazu wird zunächst ein kollisionsfreier Pfad zwischen zwei Arbeitsbereichen ermittelt, auf Basis dessen ein Übergangskorridor aufgespannt wird. Für diese beiden Arbeitsschritte ist eine Kollisionsüberprüfung notwendig. Daher wird in 4.3 zunächst auf die Kollisionsprüfung eingegangen. Abschließend wird erläutert, wie nach Ermittlung der Übergangsbereiche innerhalb dieser und der Arbeitsbereiche, sowie optional innerhalb der Störkonturen, Punkte generiert und klassifiziert ausgegeben werden.

4.1 Umgebungsrepräsentation

In dem *Pre-Processor* wird auf Basis eines Umgebungsmodells die von dem Roboter zu erfüllende Bewegungsaufgabe definiert. Aus diesem Grund ist schon bei der Wahl der Umgebungsrepräsentation darauf zu achten, dass gewisse Anforderungen erfüllt werden. Diese Anforderungen sind die Modellgenauigkeit, die Verfügbarkeit der Modelle aus gängigen CAD-Systemen, eine ausreichende MATLAB-Unterstützung der Modelle für weitere Operationen, Manipulierbarkeit der einzelnen Objekte, Speicherbedarf der Umgebungsrepräsentation sowie die Eignung für eine Kollisionserkennung. Anhand dieser Kriterien fällt die Wahl der Umgebungsrepräsentation auf einen dualen Ansatz. Eine Übersicht sowie Eigenschaften der Modelle sind in [6] gegeben.

Für die Anzeige und Interaktion innerhalb der Benutzeroberfläche des Tools wird das Begrenzungsflächenmodell [7] gewählt. Zusätzlich zu dieser Repräsentation wird für die Kollisionserkennung eine *Bounding Volume Hierarchy*

mit *Axis-Aligned Bounding Boxes* (AABBs) herangezogen [8].

Um die Verarbeitung der Daten innerhalb des *Pre-Processors* unabhängig vom verwendeten CAD-System zu gestalten, ist ein gängiger Ansatz der Export des Umgebungsmodells in ein neutrales, standardisiertes Dateiformat. Dieses kann dann wiederum von *PreSynth* eingelesen und verarbeitet werden. Als eines der ältesten und einfachsten Formate bietet sich für diese Aufgabe das Dateiformat *STL* an, was standardmäßig als Schnittstelle zwischen CAD-Programmen dient. Das Format beschreibt ausschließlich *Triangle Meshes* und besitzt einen einfachen Aufbau [9]. Vorteile dieses Dateiformats sind die große Verbreitung und Ausgereiftheit des Standards unter allen CAD-Programmen sowie die Verfügbarkeit von Funktionen in MATLAB, die mit diesem Dateiformat umgehen können. Für die Umsetzung des Synthesetools wird die Funktionssammlung *stlTools* [10] verwendet, um *STL*-Dateien einzulesen und umwandeln zu können. Nach Import der Daten können einzelne Objekte bei Bedarf neu positioniert und orientiert werden oder zusätzlich importiert werden (siehe **Bild 4**). So kann der Anwender die Umgebung intuitiv und flexibel anpassen, ohne auf einen neuen Export aus dem CAD-System angewiesen zu sein. Mittels Schaltflächen zur Interaktion können Modelle und Objekte geladen und nach Anpassung gespeichert werden. Außerdem kann die Roboterposition interaktiv vorgegeben werden, sodass diese im Verarbeitungsprozess des *Pre-Processors* mit berücksichtigt wird, was sich auf die Ermittlung der Übergangsbereiche auswirkt.

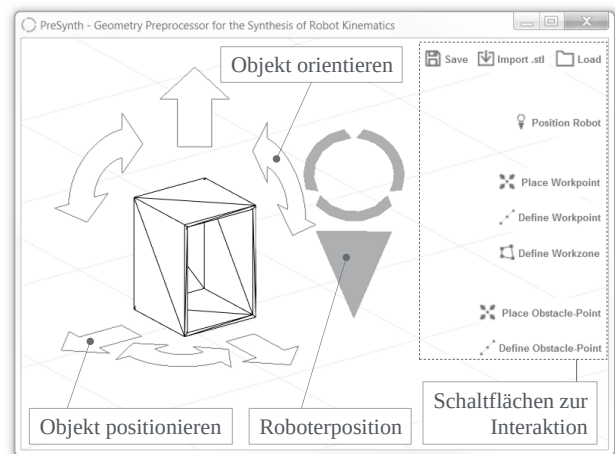


Bild 4 Visualisierung und Anpassung des Modells

4.2 Definieren der Arbeitsbereiche

Die Handhabungsaufgabe, welche von der Sonder-Roboterkinematik erfüllt werden soll, wird in *PreSynth* durch die Vorgabe von Arbeitsbereichen und Arbeitspunkten innerhalb des Umgebungsmodells definiert. Der Anwender legt fest, welche Bereiche zwangsläufig mit der Regionalstruktur erreichbar sein müssen. Auf Basis dieser Vorgaben werden im Verarbeitungsprozess Punktmengen ermittelt und im Anschluss klassifiziert ausgegeben.

Um einen Arbeitsbereich zu definieren, kann eine zusammenhängende, ebene Objektfläche des Umgebungsmodells ausgewählt werden. Diese Fläche ist ein Polygon und wird durch eine Menge von Punkten des Modells, die miteinander verbunden werden, gebildet. Alternativ können auch einzelne Objektpunkte gewählt werden, wodurch das Polygon durch den Anwender selbst definiert und aufgespannt wird (siehe **Bild 5**). Um den Arbeitsbereich aufzuspannen, wird das Polygon entlang der Flächennormalen um eine wählbare Distanz verschoben, wodurch ein sogenanntes *Sweep-Volumen* erzeugt wird [11].

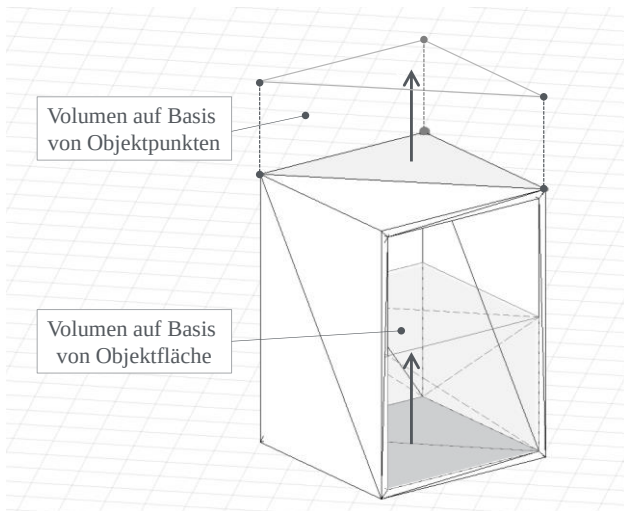


Bild 5 Erzeugung der Arbeitsbereiche als Sweep-Volumen

Neben den Arbeitsbereichen hat der Anwender auch die Möglichkeit, einzelne Arbeitspunkte vorzugeben. Dies ist besonders dann hilfreich, wenn die gesuchte Roboterkinematik bestimmte einzelne Positionen mit der Regionalstruktur auf jeden Fall erreichen können soll. Um flexibel auch zusätzliche Störkonturen anzudeuten, können optional Störkonturpunkte eingefügt und verschoben werden. Diese werden bei der Ermittlung der Übergangsbereiche berücksichtigt und können weiterhin als zusätzliche Nebenbedingungen beim Konzipieren und Entwerfen des Roboters herangezogen werden.

4.3 Ermittlung der Übergangsbereiche

Nach optionaler Anpassung des Umgebungsmodells sowie der vollständigen Vorgabe der Arbeitsbereiche und Arbeitspunkte, kann der Verarbeitungsprozess des *Pre-Processors* erfolgen. Es werden paarweise Übergangsbereiche zwischen allen Arbeitsvolumen ermittelt. Die Übergangsbereiche werden so generiert, dass sie frei von Kollisionen mit den Störkonturen des Umgebungsmodells sind. Die Motivation hinter der Berechnung dieser Übergangsbereiche liegt in der taskbasierten Kinematik-Synthese. Einerseits soll die Regionalstruktur des Roboters in allen vorgegebenen Arbeitsbereichen handhaben können, andererseits sollte berücksichtigt werden, dass sich die Kinematik von einem Arbeitsbereich in einen weiteren Arbeitsbereich bewegen kann. Es besteht außerdem die Anforderung, dass der Übergangsbereich groß genug ist, sodass auch ein Objekt definierbarer Größe von einem Arbeitsbe-

reich zu einem anderen Arbeitsbereich kollisionsfrei bewegt werden kann. Die Ermittlung der Übergangsbereiche ist unabhängig von der Roboterkinematik und die in dem Bereich generierten Punkte können optional beim Konzipieren und Entwerfen berücksichtigt werden, stellen jedoch kein notwendiges Kriterium dar.

Die Ermittlung der Übergangsbereiche ist in zwei Hauptaufgaben unterteilt. Zuerst wird ein kollisionsfreier Pfad zwischen zwei Arbeitsbereichen ermittelt. Dieser Pfad wird im anschließenden Arbeitsschritt zu einem Übergangskorridor ausgebildet, welcher das Umgebungsmodell nicht schneidet.

Damit sowohl Pfad als auch Korridor nicht mit dem Umgebungsmodell kollidieren, wird eine Kollisionsüberprüfung zwischen Punkt-zu-Punkt-Strecken und Umgebungsobjekten durchgeführt. Nachfolgend wird zunächst auf die Kollisionsprüfung und das dazu verwendete Modell eingegangen und anschließend die Pfadplanung sowie die Ausweitung des Pfades zu einem Übergangsbereich erläutert.

Kollisionsüberprüfung

Eine Kollisionsüberprüfung mit Begrenzungsflächenmodellen kann zeitaufwändig und rechenintensiv sein, da bei einem komplexen Umgebungsmodell eine Vielzahl an *Triangle Meshes* auf Kollision überprüft werden muss. Daher wird aus dem Begrenzungsflächenmodell ein Repräsentationsmodell generiert, um die Anzahl der durchzuführenden Prüfvorgänge bei der Kollisionserkennung zu minimieren. Dazu wird das Umgebungsmodell zusätzlich durch einfache Primitive approximiert, die ein schnelles Ausschließen von Teilmengen des Modells ermöglichen. Da diese einfachen Körper das komplexe Modell ganz oder teilweise umschließen, spricht man hierbei von sogenannten *Bounding Volumes*. [8]

In **Bild 6** sind verschiedene Möglichkeiten der *Bounding Volumes* visualisiert. Abgebildet sind von links nach rechts das Original, die *Bounding Sphere*, die *Oriented Bounding Box* (OBB), die *Axis-Aligned Bounding Box* (AABB) sowie ein *x-Discretely Oriented Polytope* (k-DOP). Weitere Informationen zu den *Bounding Volumes* sind in der Literatur [8] gegeben.

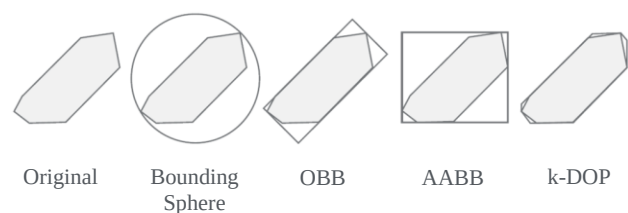


Bild 6 Bounding Volumes

Für eine effiziente Kollisionserkennung bieten sich vor allem *Bounding Volumes* in hierarchischer Struktur an, die sogenannten *Bounding Volume Hierarchies*. Dabei sind einzelne *Bounding Volumes* in weitere *Bounding Volumes* unterteilt. In **Bild 7** ist ein Beispiel in einer hierarchischen Struktur mit AABBs als *Bounding Box* mit zugehörigem Baum abgebildet. Gerade bei hochaufgelösten Modellen lassen sich durch eine solche Struktur viele Kollisi-

onsfälle schnell ausschließen und die Zahl der Kollisionsüberprüfungen wird deutlich reduziert. [8]

Für Kollisionsüberprüfungen sind allgemein *Bounding Spheres* besonders geeignet, da bei diesen Hüllkörpern die numerisch vorteilhafte Kugel-zu-Kugel Kollisionserkennung durchgeführt werden kann [8]. Dennoch wird vorerst in dem *Pre-Processor* eine *Bounding Volume Hierarchy* mit AABBs verwendet. Dies liegt einerseits an der Verfügbarkeit der Funktionsbibliothek [12] in MATLAB, mit welcher ebendiese Struktur aus einem Begrenzungsflächenmodell erzeugt werden kann. Andererseits ist die Tatsache, dass eine Kollisionsprüfung auf Basis solch einer Darstellung weniger effizient ist als bei der Darstellung mittels *Bounding Spheres*, innerhalb des *Pre-Processors* nicht von signifikantem Nachteil, da die Kinematik-Synthese offline erfolgen kann und nicht zeitkritisch ist.

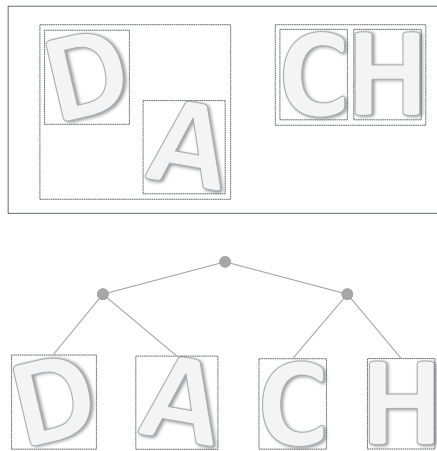


Bild 7 Bounding Volume Hierarchy mit AABBs

Durch die Verwendung der *Bounding Volume Hierarchy* mit AABBs wird daher anstatt zwischen Kugel und Kugel eine Kollisionserkennung zwischen einer Strecke und einzelnen Boxen durchgeführt. Die orientierten Eigenschaften von AABBs begünstigen dabei die Feststellung, ob eine Strecke in Kollision ist. Die Funktionsbibliothek [12] beschreibt jede einzelne Box durch die Angabe der minimalen und maximalen Achsenwerte (P_{min} und P_{max} in **Bild 8**).

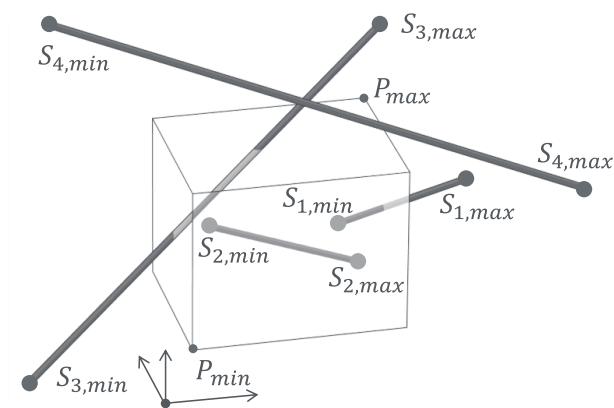


Bild 8 Strecke-Box Darstellung und Kollisionsprüfung

Anhand dieser Werte werden die Boxen im Koordinatensystem entlang der Achsen aufgespannt. Um nun eine Strecke mit Start und Endwert auf Kollision mit solch einer Box zu prüfen, kann mithilfe eines Vergleiches der Start- und Endwerte von Strecke und Box schnell eine erste Aussage getroffen werden.

Liegt nur ein Punkt der Strecke im Wertebereich, schneidet die Strecke die Oberfläche der *Bounding Box* und kollidiert mit der Box (siehe **Bild 8**, Strecke 1). Befinden sich beide Punkte innerhalb des Wertebereichs der AAB, liegt die Strecke innerhalb der Box und eine Kollision liegt vor (siehe **Bild 8**, Strecke 2). Lediglich wenn beide Punkte außerhalb des Wertebereichs liegen, kann keine exakte Aussage über den Kollisionszustand getroffen werden. Die Strecken 3 und 4 in **Bild 8** verdeutlichen dies. Obwohl beide Punkte außerhalb des Achswertebereichs der AAB liegen, schneidet eine Strecke die Box und die andere nicht. In diesem Fall werden die Dreiecksflächen des Begrenzungsflächenmodells auf Kollision überprüft.

Um nun das Gesamtmodell auf eine Kollision mit einer Strecke zu prüfen, wird die Kollisionsüberprüfung iterativ durchgeführt. Der Algorithmus startet mit der Evaluierung der *Bounding Box* auf der ersten Hierarchieebene und überprüft, ob die Strecke mit der Box kollidiert. Ist dies der Fall, wird ermittelt, ob in dieser Box bereits Dreiecksflächen des Begrenzungsflächenmodells enthalten sind. Falls Dreiecksflächen enthalten sind, werden diese einzeln auf Kollision überprüft. Liegt eine Kollision mit einer Fläche des Begrenzungsflächenmodells vor, bricht der Algorithmus ab und gibt eine Kollision aus. Falls keine Kollision mit dem Begrenzungsflächenmodell vorliegt, betrachtet der Algorithmus die *Bounding Boxes*, die der kollidierenden *Bounding Box* untergeordnet sind und beginnt erneut mit der Überprüfung der nächsten Box und der darin liegenden Flächen des Begrenzungsflächenmodells. Falls keine Boxen mehr übrig sind, gibt der Algorithmus aus, dass keine Kollision vorliegt.

Pfadplanung

Um zwei Arbeitsbereiche mit einem kollisionsfreien Pfad zu verbinden, wird für die Pfadplanung innerhalb des *Pre-Processors* ein *Rapidly-exploring Random Tree* (RRT), also ein aktiver stichprobenbasierter Algorithmus, verwendet [13,14]. In der Literatur [15] ist eine Übersicht mit entsprechenden Eigenschaften der verschiedenen Algorithmen gegeben.

Grund für die Wahl des RRT als Pfadplanungsalgorithmus ist einerseits die geeignete Zeitkomplexität sowie die einfache Integrationsmöglichkeit in MATLAB. Dazu kann die Funktionsbibliothek [16] verwendet werden.

Schon während der Pfadermittlung soll sichergestellt werden, dass auch ein Objekt mit frei wählbarer Größe kollisionsfrei durch das Umgebungsmodell von einem Arbeitsbereich zu einem weiteren bewegt werden kann. Das Handhabungsobjekt wird mit einer Kugel angenähert und der Durchmesser kann vom Anwender frei vorgegeben und an die Aufgabe angepasst werden. Damit der Pfad nicht zu nah an Störkonturen des Umgebungsmodells verläuft, wird

zu dem eigentlichen Streckenabschnitt zusätzlich überprüft, ob eine Kollisionen mit weiteren, zu dem eigentlichen Pfad parallelen Strecken auftritt. Die zusätzlichen Strecken nähern die Außenhülle der Objektkugel an. Diese Problematik sowie der Lösungsansatz sind in **Bild 9** dargestellt.

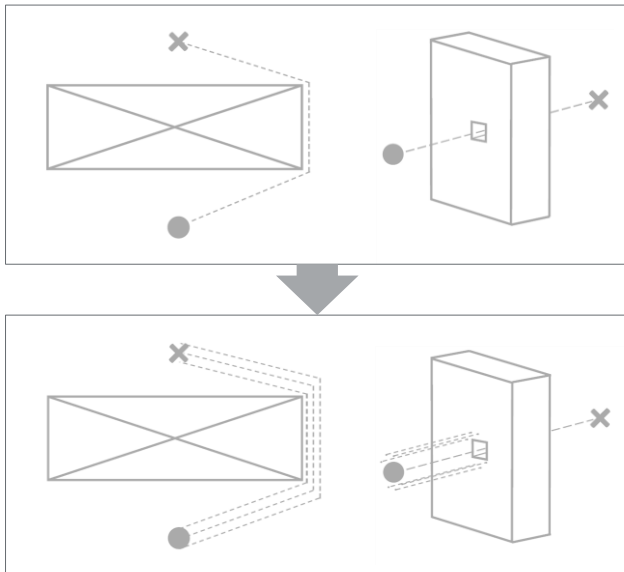


Bild 9 Pfaderweiterung für definierbare Objektgröße

Damit die Kinematik-Synthese durch die Störkontur nicht negativ beeinflusst wird, was durch einen ungünstigen Pfad passieren könnte, wird die Anordnung der Umgebungsobjekte in Bezug auf die Roboterposition mit berücksichtigt. In **Bild 10** ist ein solches Szenario veranschaulicht. Wird der kürzere Pfad 1 ermittelt, wirkt sich das negativ auf die Kinematik-Synthese aus, falls die Positionen aus dem Übergangsbereich mitberücksichtigt werden. Es besteht die Gefahr, dass beispielsweise die Gliedlängen unnötigerweise überdimensioniert werden. Um dies zu vermeiden, werden Schatten der einzelnen Objekte projiziert, die als zusätzliche Störkontur berücksichtigt werden. Die Projektion erfolgt in Abhängigkeit von der frei definierbaren Roboterposition. Auf diese Weise können ungünstige Pfade vermieden werden.

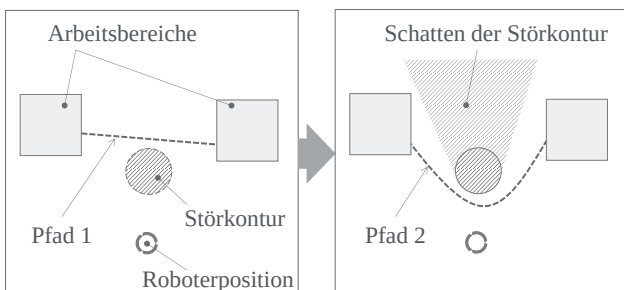


Bild 10 Anpassen der Pfadgenerierung

Übergangskorridor

Um einen Übergangsbereich auf Basis des gefundenen Pfades auszubilden, werden zuerst vom Start- und Zielvolumen alle Punkte des Begrenzungsflächenmodells von dem entsprechenden Arbeitsbereichen auf eine zum Pfad senkrechte Ebene projiziert und jeweils miteinander verbunden. Das dadurch entstehende Polygon wird entlang des Pfades verschoben und somit ein erstes Volumen aufgespannt. Damit auch das Handhabungsobjekt mit der vom Nutzer definierten Größe durch den Übergangsbereich manipuliert werden kann, wird überprüft, dass der Kreis mit maximal möglichem Durchmesser innerhalb des jeweiligen Polygons mindestens den Durchmesser aufweist, wie der Durchmesser der Kugel, welche das Handhabungsobjekt annähert. Um den Übergangsbereich weiter auszubilden, werden im nächsten Schritt Zwischenformen der ersten beiden Querschnitte erstellt und einander angenähert. Diese Querschnitte werden dann an den einzelnen Knoten des Pfades positioniert und so ausgerichtet, sodass ein geeigneter Übergangsbereich mithilfe der Start- und Zielquerschnitte und den Zwischenformen entsteht. In **Bild 11** ist ein Umgebungsmodell mit Übergangsbereich abgebildet. Der dargestellte Übergangsbereich wurde mit dem beschriebenen Verfahren in *PreSynth* ermittelt.

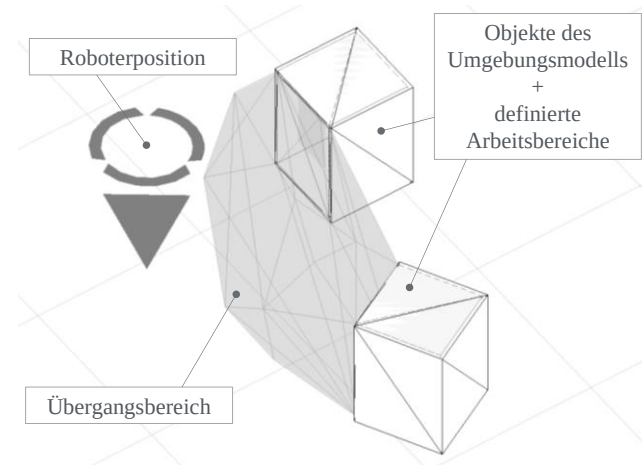


Bild 11 Visualisierung des Übergangsbereichs in *PreSynth*

4.4 Punktesampling

Aufgabe des *Pre-Processors* ist es, die Bewegungsaufgabe zu planen und die Input-Daten für den *Solver* auf Basis des Umgebungsmodells bereitzustellen. So kann eine taskbasierte Kinematik-Synthese durchgeführt werden. Daher werden im Anschluss an die Ermittlung der Übergangsbereiche Punktemengen generiert. Diese Punkte wiederum werden im *Solver* als Vorgabepositionen herangezogen. Um eine gute Abdeckung des gesamten Volumens aus Arbeitsbereichen und Übergangsbereichen sicherzustellen, empfiehlt es sich, Punkte auf den Oberflächen und innerhalb der Volumenkörper zu erzeugen. Für das Generieren dieser Punkte werden in *PreSynth* drei Sampling-Methoden verwendet. Diese sind das *Random Sampling*, *Quasi-Random Sampling* und *Discrete Sampling* [17].

Drei wünschenswerte Kriterien für die Auswahl von Punkten in einem Raum sind Uniformität, eine Raster-Struktur und die inkrementelle Qualität. Uniformität steht für eine gute Abdeckung des zu Verfügung stehenden Raumes, ohne dass Punkte sich anhäufen oder Lücken in der Stichprobe entstehen. Bei Raster-Strukturen ist aufgrund der Struktur die Lage benachbarter Punkte einfach bestimmbar. Die inkrementelle Qualität bezieht sich darauf, dass auch bei einem plötzlichen Abbrechen der Auswahlsequenz die bereits generierten Punkte eine gute Abdeckung des Raumes liefern. [17]

Damit ein Einfluss der Sampling-Methode auf die Kinematik-Synthese untersucht und an die Aufgabenstellung angepasst werden kann, sind in *PreSynth* die drei genannten Verfahren implementiert und können vom Anwender gewählt werden. Je nach Sampling Methode bietet die interaktive Benutzeroberfläche Eingabemöglichkeiten für verschiedener Einstellungen. Für *Random* und *Quasi-Random Sampling* kann der Benutzer angeben, wie viele Punkte jeweils für jede Art von Volumenkörper auf der Oberfläche und im Inneren erzeugt werden sollen. Beim *Discrete Sampling* ist die vorherige Abschätzung der resultierenden Punktemenge nicht möglich, sodass der Benutzer hier das Abstandsmaß zwischen den Punkten definiert. Mittels *Discrete Sampling* werden dann gleichmäßig Punkte mit der gewählten Diskretisierung sowohl auf der Oberfläche als auch im Inneren des jeweiligen Volumenkörpers generiert.

In **Bild 12** ist ein Umgebungsmodell mit Arbeitsbereichen, dem ermittelten Übergangsbereich sowie den generierten Punkten abgebildet.

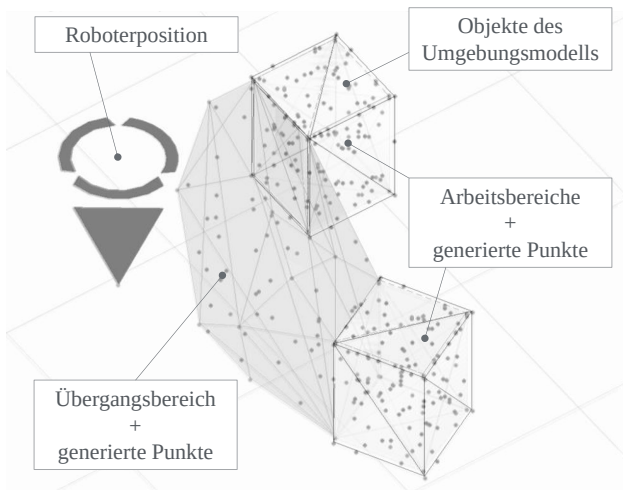


Bild 12 Visualisierung der generierten Punkte in PreSynth

4.5 Klassifizierung und Ergebnisausgabe

Die Punkte werden innerhalb der Arbeitsbereiche, Übergangsbereiche und Störkonturen generiert und kategorisiert an den *Solver* weitergegeben. So können sie als Vorgabepositionen für den weiteren Syntheseablauf verwendet werden. Außerdem werden weit von der Roboterposition entfernte Punkte sowie die am weitesten voneinander entfernten Punkte identifiziert. Diese Punkte besitzen eine

höhere Aussagekraft über die Eignung einer Kinematik für den geforderten Task als andere und können durch die Klassifizierung priorisiert behandelt werden. Daher werden die Punkte für die weitere Verarbeitung in einem MATLAB Objekt namens *Taskgeometry* strukturiert und in der Ausgabedatei zur weiteren Verarbeitung gespeichert. Die Struktur dieses Objektes ist in **Bild 13** dargestellt.

```
classdef Taskgeometry
    properties

        Prio_WorkPoints,
        Prio_Extrema,
        Prio_Workspace_in,
        Prio_Workspace_out,

        StoerK_Prio,
        StoerK_Vertices,
        StoerK_Samples_surf,
        StoerK_Samples_interior,

        ArbeitsB_Vertices,
        ArbeitsB_Samples_surf,
        ArbeitsB_Samples_interior,

        TransV_Vertices,
        TransV_Samples_surf,
        TransV_Samples_interior,

        RobotPosition,
        SamplingMethod,
        Data

    end
end
```

Bild 13 Struktur des Ausgabe-Objektes

5 Zusammenfassung und Ausblick

Dieser Beitrag beschreibt den ersten Arbeitsschritt einer Methode sowie die Umsetzung für eine taskbasierte Kinematik-Synthese von Roboter-Regionalstrukturen. Durch den *Pre-Processor* werden die Randbedingungen aus der Arbeitsumgebung sowie zusätzliche Nutzereingaben und die Roboterposition mitberücksichtigt. Als Output-Daten liefert der *Pre-Processor* klassifizierte Punktemengen innerhalb der Arbeitsbereiche und innerhalb der Übergangsbereiche zwischen den einzelnen Arbeitsbereichen. Diese können in einem nachfolgenden Arbeitsschritt der Kinematik-Synthese als Vorgabepositionen herangezogen werden. Zusätzlich werden Punktemengen innerhalb der Störkonturen erzeugt, die optional bei der weiteren Synthese mitberücksichtigt werden können. Für das generieren der Punktemengen stehen drei Sampling-Methoden zur Verfügung.

Durch die Klassifizierung und Priorisierung der Positionen kann das Lösungsfeld bei der weiteren Synthese bereits

frühzeitig eingeschränkt und somit die Effizienz gesteigert werden. Auf diese Weise ist es möglich, die notwendigen Daten für das Konzipieren und Bemessen von Roboterkinematiken auf Basis beliebiger CAD-Umgebungsmodelle flexibel bereitzustellen. Durch die Teilautomatisierung des Verfahrens wird der Anwender systematisch durch den ersten Schritt bei der Kinematik-Entwicklung geführt und somit beim Planen der Bewegungsaufgabe unterstützt.

Um die Berechnungszeit bei der Kollisionsüberprüfung zu verbessern, kann das verwendete Repräsentationsmodell sowie der Pfadplanungsalgorithmus selbst verändert werden, damit eine Kugel-zu-Kugel Kollisionsüberprüfung stattfinden kann.

Weitere Arbeiten werden sich auf den *Solver* der beschriebenen Methode für die Kinematik-Synthese beziehen. Die vom *Pre-Processor* bereitgestellten Daten werden innerhalb des *Solvers* als Vorgabepositionen verwendet, um eine aufgabenbasierte Struktur- und Maßsynthese durchzuführen.

6 Literatur

- [1] Corves, B.; Niggemann, H.: *CAD-gestützte, grafische Maßsynthese sphärischer und räumlicher Übertragungskurvengetriebe mit dem Programm CADis*. In: VDI-Getriebetagung 2004, 21./22. September 2004 Fulda. VDI-Gesellschaft Entwicklung Konstruktion Vertrieb. ISBN 3-18-091845-4.
- [2] VDI: VDI 2221. *Methodik zum Entwickeln und Konstruieren technischer Systeme und Produkte*. Verein Deutscher Ingenieure. Berlin: Beuth Verlag GmbH, 1993.
- [3] Abdul-Sater, K.: *Task-Based, Computer-Aided Kinematic Design of Spherically Constrained Kinematic Chains*. München: Technische Universität München, Lehrstuhl für Mikrotechnik und Medizintechnik, 2016.
- [4] Fischer, M.: *Anwendungsspezifische Kinematikentwicklung unter Einsatz modularer Prototypen*. München: Technischen Universität München, Lehrstuhl für Mikrotechnik und Medizintechnik, 2010.
- [5] Ruokonen, A.: *Environment- and task-driven tool for selecting industrial robots*. Tampere: Tampere University of Technology, Department of Automation Science and Engineering, 2016.
- [6] Schinko, C. et al.: *State-of-the-Art Overview on 3D Model Representations and Transformations in the Context of Computer-Aided Design*. International Journal on Advances in Software. (2017) 10, ISBN 978-1-61208-548-7, S.10-15.
- [7] Schroeder, W. et al.: *Visualization Toolkit: An Object-Oriented Approach to 3D Graphics*, New York NY: Kitware, ISBN 1-930934-19-X, 2006.
- [8] Weller, R.: *New Geometric Data Structures for Collision Detection and Haptics*, Cham, Heidelberg, New York, Dordrecht, London: Springer International Publishing, ISBN 978-3-319-01019-9, 2013.
- [9] Rumpf, D.: *Synthesizing Geometries for 21st Century Electromagnetics, Interfacing MATLAB with CAD*, El Paso, Texas, USA, Department of Electrical and Computer Engineering, 2016.
- [10] Micó, P.: *stlTools* <https://de.mathworks.com/matlabcentral/fileexchange/51200-stltools> (zuletzt geprüft am 03.02.2019).
- [11] Requicha, A.: *Representations for Rigid Solids*. ACM Computing Surveys. (1980) 4, DOI 10.1145/356827.356833, S. 437–464.
- [12] Engwirda, D.: *AABB-Tree* <https://github.com/dengwirda/aabb-tree> (zuletzt geprüft am 03.02.2019).
- [13] Karaman, S.; Frazzoli, E.: *Incremental Sampling-based Algorithms for Optimal Motion Planning*. In: Robotics: Science and Systems 2010. 27.-30. Juni 2010 Zaragoza, Spain.
- [14] Karaman, S.; Frazzoli, E.: *Sampling-based algorithms for optimal motion planning*. The International Journal of Robotics Research. (2011) 7, DOI 10.1177/0278364911406761, S. 846–894.
- [15] Yang, L. et al.: *Survey of Robot 3D Path Planning Algorithms*. Journal of Control Science and Engineering. (2016) 1, DOI 10.1155/2016/7426913. S. 1–22.
- [16] Vemprala, S. H.: *2D/3D RRT* algorithm* <https://de.mathworks.com/matlabcentral/fileexchange/60993-2d-3d-rrt--algorithm> (zuletzt geprüft am 03.02.2019).
- [17] Lindemann, S. R.; LaValle, S. M.: *Incremental low-discrepancy lattice methods for motion planning*. In: IEEE International Conference on Robotics and Automation 2003. 14.-19. September 2003, Taipei, Taiwan. ISBN 0-7803-7736-2, S. 2920–2927.

Online-Optimierung von Geschwindigkeitsprofilen entlang vorgegebener Bahnen unter Berücksichtigung von Achsruckbeschränkungen

Online-optimization of path-restricted feedrate profiles under consideration of axis jerk restrictions

Justus Kopp, Bosch Rexroth AG, Priv. Univ. UMIT (Hall/Tirol), justus.kopp@boschrexroth.de

Dr. Bertold Bader, Bosch Rexroth AG, bertold.bader@boschrexroth.de

Univ.-Prof. Dr.-Ing. Frank Woittennek, Priv. Univ. UMIT (Hall/Tirol), frank.woittennek@umit.at

Kurzfassung

Der vorliegende Beitrag präsentiert einen Algorithmus zur Berechnung eines näherungsweise zeitoptimalen Geschwindigkeitsprofils entlang einer vorgegebenen Bahn. Es werden Beschränkungen der Geschwindigkeit entlang der Bahn und Beschränkungen der an der Bewegung beteiligten Achsen bis zur dritten Ableitung, dem Ruck, betrachtet. Der Algorithmus baut auf der Struktur einer Werkzeug- oder Handlingmaschinensteuerung auf, die eine Werkzeugspitze entlang einer vorgegeben oder auch intern modifizierten Bahn verfährt. Dabei verwendet der Algorithmus die intern berechneten Zusammenhänge zwischen Achs- und Bahnkoordinaten. Die Anzahl der Maschinenachsen, die an der Bewegung beteiligt sind, ist theoretisch beliebig.

Für eine schnelle, näherungsweise zeitoptimale Berechnung des Profils wird die vorgegebene Raumkurve gezielt in mehrere Teilabschnitte zerlegt, die Übergangsbedingungen gewählt und in jedem dieser Abschnitte ein Geschwindigkeitsprofil mittels einer Vorwärts-/Rückwärtsrechnung konstruiert. Dabei erweitert der Beitrag den Lösungsansatz aus [8]. Im Mittelpunkt stehen die Definition des Geschwindigkeitsprofils, als einmal stetig differenzierbarer Spline mit stückweise konstantem Ruck, die Aufteilung der Bahn in einzelne Berechnungsabschnitte und die Simulationsergebnisse mit Daten einer Fünfachsbearbeitung. Ein Nachweis zur Optimalität und online-Einsetzbarkeit der Lösung wird nicht angegeben.

Abstract

This paper presents an algorithm for the calculation of an approximately time-optimal velocity profile along a given path. Restrictions of the velocity along this path and restrictions of the axes involved in the motion up to the third derivative, the jerk, are considered. The algorithm is based on the structure of a tool or handling machine control with a theoretically arbitrary number of machine axes, which moves a tool tip along a given or internally modified path. The algorithm uses their internally calculated relationships between axis and path coordinates. For a fast and approximately time-optimal calculation of the profile, the given path is specifically divided into several sub-sections, transition conditions are selected and a velocity profile is constructed in each of these sections by means of a forward/backward calculation. The contribution extends the solution approach from [8]. In focus is the definition of the velocity profile, as C^1 spline with piecewise constant jerk, the splitting into parts for the calculation and the simulation results with data of a five-axis machining. A proof of the optimality and online usability of the solution is not given.

1 Einleitung

Im Bereich der Werkzeug- und Handlingmaschinen wird das Trio aus Geschwindigkeit, Genauigkeit und Flexibilität immer wichtiger. So muss auf Grund von geforderten minimalen Taktzeiten die größtmögliche Bearbeitungsgeschwindigkeit gefahren werden, ohne dabei die Oberflächengüte des Werkstücks zu verschlechtern. Weiterhin kommt hinzu, dass Maschinen gleicher Größe immer leichter und weniger starr konstruiert werden. Durch die niedrigeren Eigenfrequenzen ist es deshalb auch eine zunehmende Herausforderung die Maschine stabil und Schwingungsarm zu bewegen. Aus diesen Gründen ist die Berücksichtigung des Achsruckes, der zeitlichen Ableitung der Achsbeschleunigung, bei der Berechnung von Verfahrgeschwindigkeiten bereits Stand der Technik. Dies führt zu längeren Taktzeiten, verringert jedoch die Schwingungen im System. Da teilweise die Achsbeschleunigungen somit

weniger stark beschränkt werden müssen, führt dies sogar zu kürzeren Taktzeiten. Weiterhin darf die Variabilität nicht vernachlässigt werden. So ist es weiterhin wichtig, die Bahngeschwindigkeit online und während der Bewegung zu berechnen, um auf unvorhersehbare Stelleingriffe aus der SPS oder vom Bediener reagieren zu können.

Die Literatur zeigt bereits eine Reihe von Lösungsvorschlägen für die Problemklasse auf. Anfänglich wurden Lösungsvorschläge geliefert, die lediglich Nebenbedingen bis hin zur Beschleunigung betrachten [4, 5, 9]. Wie oben beschrieben, wird zunehmend auch die Betrachtung von Ruckbeschränkungen der Achsen gefordert. Diese verändern zwar kaum die Problemstellung, erschweren jedoch die Berechnung der Bahngeschwindigkeit um ein Vielfaches. Die Literatur [2, 6, 8, 11] liefert trotzdem vielversprechende Ansätze. In der Industrie finden diese Verfahren bisher eher selten Anwendung, da sie entweder von vornherein für eine offline-Berechnung ausgelegt [11] oder zu-

mindest zu aufwändig für die online-Berechnung in einer mehrkanaligen Werkzeugmaschinensteuerung (CNC) sind [3]. Lösungen, die sich nur auf ein bestimmtes mechanisches System beziehen [7], scheiden für den allgemeinen Einsatz in einer CNC ebenfalls aus.

Der Beitrag greift die genannte Herausforderung auf und zeigt eine Lösung für beliebige Maschinenkinematiken und umgeht die Nachteile, etwa den hohen Berechnungsaufwand, bisheriger Verfahren. Dabei liegt der Fokus vorrangig auf der Berechnung eines näherungsweise zeitoptimalen Geschwindigkeitsprofils entlang einer fest definierten, beliebigen Raumkurve. Der Algorithmus des Beitrags ist so entworfen, dass er online, in der Bahnvorbereitung einer CNC, das Geschwindigkeitsprofil der nahen Zukunft berechnet, welches im Interpolatortakt der CNC nur noch abgefahren wird.

Der Aufbau des Beitrags ist wie folgt: Im zweiten Abschnitt werden u. a. die Aufgabenstellung, die Bahn, die Nebenbedingungen und das Geschwindigkeitsprofil näher definiert. Darauf aufbauend, beschreibt Abschnitt drei den eigentlichen Algorithmus zur Bestimmung des Geschwindigkeitsprofils. Abschnitt vier liefert die Ergebnisse des Algorithmus für ein Beispielpogramm aus einer CNC.

2 Problembeschreibung

Das Problem der zeitminimalen Optimierung der Vor- und Schubgeschwindigkeit oder auch Bahngeschwindigkeit wird häufig in der Literatur diskutiert. Das Ziel dabei ist es, die Spitze einer Werkzeug- oder Handlingmaschine auf einer festen Bahn möglichst schnell zu verfahren. Die zeitoptimale Trajektorie, das Geschwindigkeitsprofil, muss dabei so berechnet werden, dass sie keine der definierten Systembeschränkungen verletzt. Eine Reihe an Nebenbedingungen, die sowohl die Bahn, als auch die beteiligten Achsen beschränken sind in [5] beschrieben.

Nachfolgend wird eine allgemeine Werkzeug- oder Handlingmaschine mit einer beliebigen Anzahl an Achsen A_n $n \in \mathbb{N}$ betrachtet. Eine beliebige Kinematik spannt dabei einen Raum mit maximal sechs Koordinaten auf: drei Raumkoordinaten x, y, z und drei Orientierungskordinaten φ, ϑ, ψ . Bei modernen Werkzeugmaschinen wird häufig eine Fünfachsbearbeitung verwendet, mit den fünf beliebigen Achsen $\mathbf{A} = (A_1 \ A_2 \ A_3 \ A_4 \ A_5)^T$ und einem rotationssymmetrischen Werkzeug, das sich relativ zum Werkstück durch fünf Koordinaten $\mathbf{P} = (x \ y \ z \ \varphi \ \vartheta)^T$ beschreiben lässt. Im Abschnitt 4 wird ein Beispiel einer solchen Fünfachsbearbeitung vorgestellt.

In dem Beitrag werden sowohl Beschränkungen der Bahn, als auch der Achsdynamik berücksichtigt. Beschränkungen, die sich aus den Fertigungsprozessen ergeben oder von einem Bediener online vorgegeben werden (F_{prog}), beziehen sich häufig auf die Geschwindigkeit der Bahn $v(s)$ selbst und werden von allen gängigen CNCs berücksichtigt. Anders stellt es sich bei den Beschränkungen der Achsen dar. Da der Zusammenhang zwischen Achs- und Raumkoordinaten als nahezu beliebig und nichtlinear anzunehmen ist, werden diese in vielen CNCs lediglich ap-

proximativ in Bahnbeschränkungen überführt. Diese Abschätzungen führen teilweise zu einer Verletzung der Achsbeschränkungen oder zu einem unnötigen Absenken der Bahngeschwindigkeit. Für die exakte Betrachtung dieser Beschränkungen wird im Folgenden zuerst der schon häufig verwendete Begriff der Bahn näher definiert.

2.1 Bahndefinition und Beschränkungen

Für Werkzeugmaschinen werden häufig durch den Bediener fest definierte geometrische Bahnen $s \mapsto \mathbf{P}(s)$ im Raum mit dem Bahnparameter s verwendet. Teilweise kann dieser Parameter als Bogenlänge $s \in [0, S]$ interpretiert werden. S entspricht dabei der Gesamtlänge der Bahn. Auch in der Robotik/Handling existiert ein solcher Bahnbegriff. Anders als bei der Werkzeugbearbeitung, spielt die exakte Einhaltung der Bahn dort jedoch eher eine untergeordnete Rolle.

Ziel des Beitrags ist es, die zeitliche Ableitung des Bahnparameters s , die Bahngeschwindigkeit $t \mapsto \dot{s}(t) = v(t)$, unter Berücksichtigung der Bahnbeschleunigung $\ddot{s}(t) = a(t)$ und des Bahnruckes $\ddot{s}(t) = j(t)$ zu berechnen. Dabei sollen die drei Ableitungen nicht durch die Zeit t , sondern durch den Bahnparameter s selbst parametrisiert sein $t \mapsto s(t) \mapsto \{v(s), a(s), j(s)\}$.

Die CNC berechnet aus dem aktuellen Bahnparameter und dem aktuell geltenden Geometrieelement¹ eine Raumkoordinate $\mathbf{P}$ (Position und Orientierung im Raum). Aus dieser Koordinate werden anschließend, abhängig von der Kinematik der Werkzeug- oder Handlingmaschine, die Positionen der beteiligten n Achsen² $\mathbf{A} = \{A_1, \dots, A_k, \dots, A_n\}$ berechnet. Typischerweise werden bei Werkzeugmaschinen für die Achsen nur konstante Beschränkungen für Position $\{A_k^{\min}, A_k^{\max}\}$, Geschwindigkeit $\{\dot{A}_k^{\max}\}$, Beschleunigung $\{\ddot{A}_k^{\min}, \ddot{A}_k^{\max}\}$ ³ und Ruck $\{\ddot{\ddot{A}}_k^{\max}\}$ parametrisiert. In der Praxis werden die Achsruckbeschränkungen häufig nur empirisch bestimmt, weshalb eine exakte Einhaltung dieser Beschränkungen nicht immer erforderlich ist.

Bei der Berechnung des Geschwindigkeitsprofils im nächsten Abschnitt werden diese Achsbeschränkungen berücksichtigt. Dafür wird ein Zusammenhang zwischen den Bahn- und Achskoordinaten $\mathbf{A}(s(t))$ bis zur dritten Ableitung, dem Achsruck $\ddot{\ddot{\mathbf{A}}}(s(t))$, benötigt. Durch Ableiten der Achskoordinaten nach der Zeit t und Anwenden der Kettenregel ergibt sich:

$$\begin{aligned} \frac{d\mathbf{A}(s(t))}{dt} &= \frac{d\mathbf{A}}{ds} \frac{ds}{dt} \\ \frac{d^2\mathbf{A}(s(t))}{dt^2} &= \frac{d^2\mathbf{A}}{ds^2} \left(\frac{ds}{dt}\right)^2 + \frac{d\mathbf{A}}{ds} \frac{d^2s}{dt^2} \\ \frac{d^3\mathbf{A}(s(t))}{dt^3} &= \frac{d^3\mathbf{A}}{ds^3} \left(\frac{ds}{dt}\right)^3 + 3 \frac{d^2\mathbf{A}}{ds^2} \frac{ds}{dt} \frac{d^2s}{dt^2} + \frac{d\mathbf{A}}{ds} \frac{d^3s}{dt^3}. \end{aligned}$$

¹Bei der Werkzeugbearbeitung sind gängige Geometrielemente Gerade, Helix und Spline.

²Linear- oder rotatorische Achsen

³In der Regel gilt: $\ddot{\ddot{A}}_k^{\min} = -\ddot{\ddot{A}}_k^{\max}$

Vereinfachte Darstellung:

$$\dot{\mathbf{A}}(s(t)) = \mathbf{A}'(s) v(s) \quad (1a)$$

$$\ddot{\mathbf{A}}(s(t)) = \mathbf{A}''(s) v(s)^2 + \mathbf{A}'(s) a(s) \quad (1b)$$

$$\ddot{\mathbf{A}}(s(t)) = \mathbf{A}'''(s) v(s)^3 + 3\mathbf{A}''(s) v(s) a(s) + \mathbf{A}'(s) j(s) \quad (1c)$$

Die Ableitungen der Achskoordinaten nach dem Bahnparameter s sind von der geometrischen und kinematischen Kette zwischen s und $\mathbf{A}$ abhängig und werden für die folgenden Betrachtungen als beliebig und nichtlinear angenommen.

2.2 Abtastung der Bahn

Für die späteren Berechnungen wird die Bahn $t \mapsto s(t)$ an diskreten Punkten s_i ausgewertet. Dabei ist eine konstante Schrittweite nicht sinnvoll. Bei starker Krümmung wird die Bahn feiner abgetastet, siehe Abb. 1. Die Orientierung, die ebenfalls Auswirkungen auf die Abtastung haben kann, ist in Abb. 1 nicht aufgetragen.

An jedem Stützpunkt s_i werden die benötigten Ableitungen der Achskoordinaten $\mathbf{A}'(s_i)$ aus dem Bahnparametrisierungsprofil

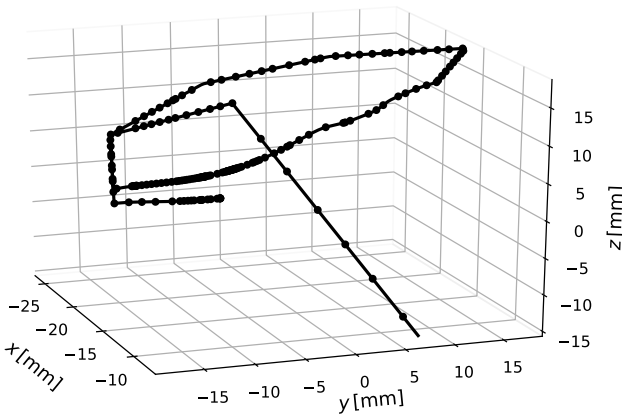


Abbildung 1 Beispiel einer beliebigen Abtastung der Schrittweite entlang der Bahn im Raum

2.3 Definition der Bahngeschwindigkeit

Die Bahngeschwindigkeit ist bisher nur an den Stützstellen s_i definiert. Für das Abfahren der Bahn werden auch Zwi-

schwerte der Bahngeschwindigkeit benötigt, die das sogenannte Bahngeschwindigkeitsprofil ergeben. Dieses Profil ist über den Bahnparameter s oder über der Zeit t parametrisiert und als einmal stetig differenzierbarer (C^1) Spline in den Gleichungen (2) definiert. Der Ruck j ist zwischen zwei Stützpunkten s_i, s_{i+1} konstant. Die weiteren Größen resultieren aus dessen Integration über der Zeit t . Bei einer Parametrisierung über s muss bei jeder Auswertung ein kubisches Gleichungssystem gelöst werden, um die aktuelle Zeit T zu berechnen.

$$\begin{aligned} j(s) &= j_i \\ a(s) &= a_i + j_i \cdot T \\ v(s) &= v_i + a_i \cdot T + \frac{1}{2} \cdot j_i \cdot T^2 \\ t(s) &= t_i + T \end{aligned} \quad (2)$$

mit $0 = (s - s_i) - v_i \cdot T - \frac{1}{2} \cdot a_i \cdot T^2 - \frac{1}{6} \cdot j_i \cdot T^3$
 $\forall s \in (s_i, s_{i+1}]$

Für die spätere Berechnung des Geschwindigkeitsprofils muss die Bahnbeschleunigung a_{i+1} und der Bahnruck j_i bei gegebenen s_i, s_{i+1}, v_i, a_i durch die Bahngeschwindigkeit v_{i+1} parametrisiert werden. Die Gln. (3) zeigen die Parametrisierung unter der Voraussetzung, dass Werte der vorherigen Stützstelle bekannt sind, dann gilt: $p = i, p_+ = i + 1$ (Vorzeichen in den Gln. + bei $\pm$). Sind hingegen die Werte der darauffolgenden Stützstelle bekannt, gibt $p = i + 1, p_+ = i$ (Vorzeichen in den Gln. - bei $\pm$).

$$\begin{aligned} T_p(v_{p_+}) &= \frac{\mp(2v_p + v_{p_+}) \pm \sqrt{(2v_p + v_{p_+})^2 + 6a_p(s_{p_+} - s_p)}}{a_p} \\ j_p(v_{p_+}) &= 2 \frac{(v_{p_+} - v_p) \mp a_p T_p(v_{p_+})}{T_p^2(v_{p_+})} \\ a_{p_+}(v_{p_+}) &= a_p \pm j_p(v_{p_+}) T_p(v_{p_+}) \end{aligned} \quad (3)$$

Diese Definition unterscheidet sich von den üblichen Lösungsansätzen der Literatur und hat zwei entscheidende Vorteile. Einerseits resultiert ein stetiger Verlauf der Achsbeschleunigungen aus beschränkten Achsruckwerten. Andererseits lässt sich das zeitparametrisierte Geschwindigkeitsprofil sehr gut in dem Interpolatortakt einer CNC verwenden. Ein großer Nachteil dieser Definition ist, dass das Profil nicht lokal definiert ist, sondern von allen Stützpunkten der Vergangenheit abhängt. Dies äußert sich vor allem bei der numerischen Optimierung, muss aber auch bei

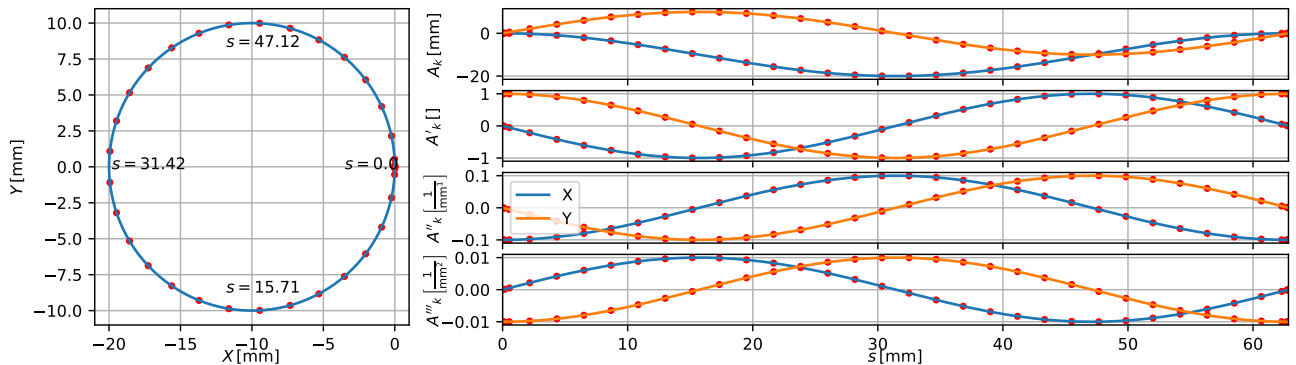


Abbildung 2 Bahn und Geometriedefinition an dem analytischen Beispiel „Kreis in der x-y-Ebene“.

dem später beschriebenen Zusammenfügen des Vorwärts-/Rückwärtszweiges berücksichtigt werden.

2.4 Anwendung am Beispiel Kreis

Die Definitionen werden zunächst an einem einfachen Beispiel erläutert. In diesem spannen zwei orthogonal angeordnete Linearachsen $\mathbf{A} = (X, Y)^T$ eine Ebene mit den orthogonalen Koordinaten x und y auf. Bei dieser Konstellation ist keine geometrische oder kinematische Transformation aktiv, sodass die Linearachsen X, Y mit den Koordinaten der Ebene x, y gleichgesetzt werden können. In dieser Ebene wird jetzt ausgehend von Nullpunkt $\mathbf{A} = \mathbf{0}$ ein Kreis im mathematisch negativen Sinn mit einem Radius $R = 10\text{ mm}$ abgefahren. Der Zusammenhang zwischen Bahn und Achskoordinaten lässt sich geschlossen angeben und lautet:

$$\begin{aligned} \mathbf{A}(s) &= (R(\cos(s/R) - 1) \quad -R \sin(s/R))^T \\ \mathbf{A}'(s) &= (-\sin(s/R) \quad -\cos(s/R))^T \\ \mathbf{A}''(s) &= (-1/R \cos(s/R) \quad 1/R \sin(s/R))^T \\ \mathbf{A}'''(s) &= (1/R^2 \sin(s/R) \quad 1/R^2 \cos(s/R))^T \end{aligned} \quad (4)$$

Die Bahn und der Zusammenhang zwischen dem Bahnparameter s und den Achskoordinaten $\mathbf{A}$ aus den Gln. (4) ist in der Abb. 2 dargestellt. Bis auf den Anfangs- und Endbereich wird hier mit einer konstanten Schrittweite auf der Bahn abgetastet.

3 Berechnung des Geschwindigkeitsprofils

Algorithmen für die näherungsweise zeitoptimale Berechnung des Geschwindigkeitsprofils in der Literatur [2, 4] nutzen sehr ähnliche Herangehensweisen. Diese berechnen, ausgehend von einem gültigen Startpunkt die Geschwindigkeit auf der nächsten Stützstelle unter Beachtung der Achsbeschränkungen und Gl. (1). Je nach Verfahren wird dabei teilweise in die Zukunft geschaut oder Beschränkungen anfangs ignoriert, um nicht in Sackgassen zu laufen.

Der Algorithmus des Beitrags ähnelt denen der Literatur. Er basiert hauptsächlich auf der schrittweise/iterativen Berechnung aus [8]. Dies bedeutet, es werden auf der Bahn markante Punkte gesucht, an denen die Bahngeschwindigkeit vorgegeben werden kann. Darauf aufbauend wird ein Zweig vorwärts von einem bekannten Punkt s_i (Startpunkt) und ein zweiter Zweig rückwärts von dem nächsten bekannten Punkt s_p (Endpunkt) mit $p > i$ berechnet. Sobald beide Zweige zusammenstoßen, werden sie verbunden: der Endpunkt wird zum Startpunkt und der nächste bekannte Punkt zum neuen Endpunkt. Von diesen Punkten wird wieder ein Vorwärts- und ein Rückwärtszweig berechnet, usw.. Der genaue Algorithmus zur Berechnung des Geschwindigkeitsprofils wird durch den Programmablaufplan in Abb. 3 beschrieben. Die Variablen i_{fwd} und i_{bwd} beschreiben den aktuellen Index im Vorwärts- (fwd) bzw. Rückwärtszweig (bwd). Der Index i_+ beschreibt die aktuell zu untersuchende Stützstelle und i die vorherige, abhängig davon, ob man sich aktuell im Vorwärts- oder Rückwärts-

zweig befindet. Alle Indizes beschreiben eine Stützstelle aus der Menge wie in Abschnitt 2.2 definiert. Es werden keine Stützstellen eingefügt oder bestehende gelöscht.

Der Parameter $\alpha \in [0, 1]$ beschreibt wie sich die Bahngeschwindigkeit v_i an der Stützstelle i aus dem berechneten Geschwindigkeitsintervall $[v_i^{\min}, v_i^{\max}]$ zusammensetzt. Da bei $\alpha \in \{0, 1\}$ teilweise nur ein Springen zwischen maximalem und minimalem Ruck erkennbar ist, bietet es sich vor allem bei einer großen Schrittweite an die Definitionsmenge von α feiner zu unterteilen. Muss auf einem Berechnungszweig jedoch zurückgegangen werden (Ablaufplan: *Gehe zurück...*), können die Iterationen exponentiell von der Anzahl der Unterteilungen abhängen. Im Abschnitt 4 gilt: $\alpha \in \{0, 0.25, 0.5, 0.75, 1\}$.

Die grau hinterlegten Blöcke von Abb. 3 werden in den folgenden Abschnitten noch einmal näher erläutert.

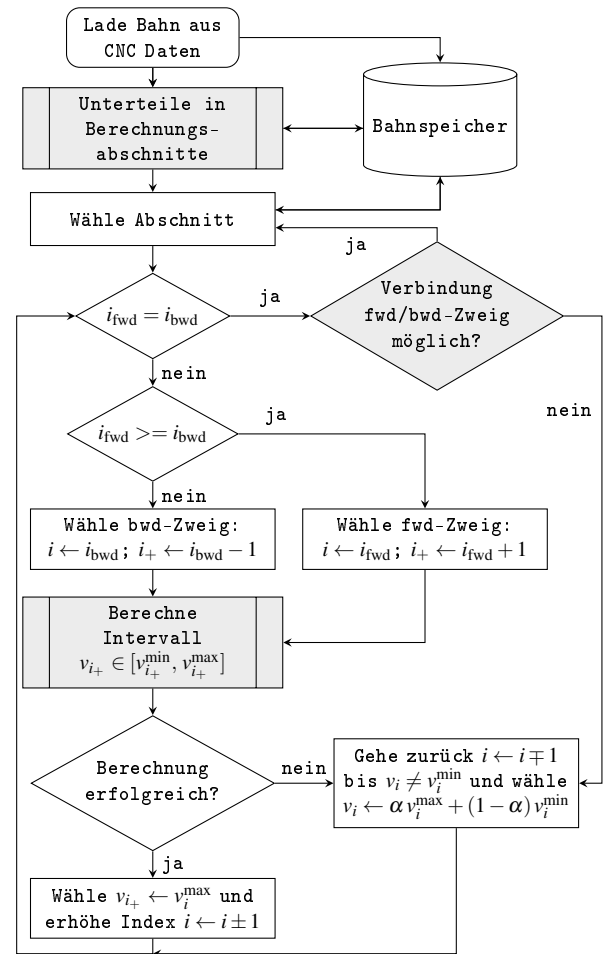


Abbildung 3 Ablaufplan des Algorithmus

3.1 Unterteilung in Abschnitte

Für die Berechnung des Geschwindigkeitsprofils mittels des Vorwärts-/Rückwärts-Algorithmus wird nachfolgend eine obere Grenze für die Bahngeschwindigkeit aus den Modellgleichungen (1) und Achsbeschränkungen geschätzt. Der Ansatz ist ebenfalls in [1] beschrieben und wird in Folgenden erweitert. Bei verschwindender Bahnbeschleunigung a und Vernachlässigung des Bahnruckes j

vereinfachen sich die Modellgleichungen aus (1) zu:

$$\begin{aligned}\dot{\mathbf{A}}(s(t)) &= \mathbf{A}'(s)v(t) \in [\dot{\mathbf{A}}_{\min}, \dot{\mathbf{A}}_{\max}] \\ \ddot{\mathbf{A}}(s(t)) &= \mathbf{A}''(s)v^2(t) \in [\ddot{\mathbf{A}}_{\min}, \ddot{\mathbf{A}}_{\max}] \\ \dddot{\mathbf{A}}(s(t)) &= \mathbf{A}'''(s)v^3(t) \in [\dddot{\mathbf{A}}_{\min}, \dddot{\mathbf{A}}_{\max}].\end{aligned}\quad (5)$$

Die obere Grenze für die Bahngeschwindigkeit $s \mapsto \bar{v}(s)$ lautet somit:

$$\bar{v}(s) = \min \left\{ F_{\text{prog}}(s), \frac{|\mathbf{A}'(s)|}{\dot{\mathbf{A}}_{\max}}, \sqrt{\frac{|\mathbf{A}''(s)|}{\ddot{\mathbf{A}}_{\max}}}, \sqrt[3]{\frac{|\mathbf{A}'''(s)|}{\dddot{\mathbf{A}}_{\max}}} \right\} \quad (6)$$

Nach den Gln. (1) ist es theoretisch möglich, dass die Bahngeschwindigkeit bei einer Bahnbeschleunigung $a > 0$ auch größer als diese obere Grenze werden kann, ohne die Achsbeschränkungen zu verletzen. In dem Fall, wo die Bahngeschwindigkeit oberhalb dieser Grenze wieder verringert werden muss, durchfährt die Beschleunigung einen Nulldurchgang. Dies hätte dann wiederum die Folge der Verletzung einer Achsbeschränkung, da dort $v > \bar{v}$ und $a = 0$. Demzufolge ist die Annahme einer verschwindenden Bahnbeschleunigung a in den Gln. (5) zulässig.

Die Vernachlässigung des Bahnruckes an der oberen Grenze ist nicht ohne Einschränkungen zulässig. Bei einer Bahngeschwindigkeit $v = \bar{v}$ und Bahnbeschleunigung $a = 0$ muss der Bahnruck auf einen Wert ungleich Null springen, was eine Verletzung des Achsruckes zur Folge hat. Ein Bahnruck $j = 0$ ist nur an Minima der oberen Grenze zulässig, da eine konstante Bahngeschwindigkeit links und rechts des Minimums bereits unterhalb der oberen Grenze liegt. In allen anderen Fällen verletzt eine konstante Bahngeschwindigkeit zu einer Seite hin eine oder mehrere Achsbeschränkungen. Deshalb wird dort bei $v = \bar{v}$ eine Verletzung des Achsruckes akzeptiert oder es muss gelten $v < \bar{v}$. Unter der Annahme einer leichten Verletzung des Achsruckes und unter der Voraussicht, dass die obere Grenze im Späteren weiter eingeschränkt wird, lautet das Lösungsgebiet der Bahngeschwindigkeit an jedem Stützpunkt s_i : $v_i \in [0, \bar{v}(s_i)] = [v_i^{\min}, v_i^{\max}]$. Die schwarze Kurve im Diagramm (rechts oben) in Abb. 4 zeigt diese obere Begrenzung für das Beispiel Kreis.

An den lokalen Minima der oberen Grenze der Bahngeschwindigkeit $\bar{v}(s_i)$ wird die Bahn in Berechnungsschnitte unterteilt. Somit wird sichergestellt, dass bei der Vorwärts-/Rückwärtsberechnung der Vorwärtzweig ausschließlich Bahnbeschleunigungen $a_{i\text{fwd}} \geq 0$ und der Rückwärtzweig $a_{i\text{bwd}} \leq 0$ enthält. Startet die Berechnung eines Vorwärts- oder Rückwärtzweiges auf diesem Minimum, ist bei $v = \bar{v}$ und $a = 0$ ein Bahnruck $j = 0$ zulässig.

3.2 Berechnung eines Schrittes im Vorwärts- oder Rückwärtzweig

Nach dem Algorithmus in Abb. 3 wird in jedem Iterationsschritt entschieden, ob diese nächste Berechnung im Vorwärts- oder Rückwärtzweig berechnet wird. Dabei wird eine aktuelle Stützstelle i_+ und die dazu gehörende vorherige Stelle i bestimmt. Abhängig davon wird das In-

itialintervall $v_{i_+} \in [0, v_{\max}(s_{i_+})]$ solange eingeschränkt, bis es allen Nebenbedingungen (7) genügt. Da die Beschränkungen (7a), (7b) bereits bei der Berechnung von $\bar{v}$ vollständig betrachtet und in dem Initialintervall berücksichtigt sind, müssen diese nicht mehr explizit überprüft werden. Da der Bahnruck über das gesamte Intervall $[s_i, s_{i_+}]$ konstant ist, wird in Gl. (7d) der Bahnruck j_i verwendet.

$$v_{i_+} \in [0, F_{\text{prog}}] \quad (7a)$$

$$|A'_k v_{i_+}| = |\dot{A}_k| \in [0, \dot{A}_k^{\max}] \quad (7b)$$

$$A''_k v_{i_+}^2 + A'_k a_{i_+} = \ddot{A}_k \in [\ddot{A}_k^{\min}, \ddot{A}_k^{\max}] \quad (7c)$$

$$|A'''_k v_{i_+}^3 + 3A''_k v_{i_+} a_{i_+} + A'_k j_i| = |\dddot{A}_k| \in [0, \dddot{A}_k^{\max}] \quad (7d)$$

Je nachdem, in welcher Reihenfolge die Nebenbedingungen (7c) und (7d) betrachtet werden, muss das Intervall unterschiedlich oft neu berechnet werden. Auch bietet es sich an nur verletzte Nebenbedingungen zu betrachten. Da der Ruck das Geschwindigkeitsprofil am ehesten beschränkt, werden zuerst alle Bedingungen des Ruckes (7d) überprüft. Die verletzte Beschränkung mit dem größten Betrag bestimmt das neue Intervall.

Dafür muss die Bahnbeschleunigung a_{i_+} und der Bahnruck j_i in der Gl. (7d) mit Hilfe der Gln. (3) aus v_{i_+} berechnet werden. Dabei ist zu beachten, dass im Vorwärtzweig (+) bzw. Rückwärtzweig (−) bei $(\pm)$ in den Gln. (3) steht. Unter Verwendung von Gl. (7d) und (3) ist der Achsruck $\ddot{A}_k(v_{i_+})$ somit nur noch von der gesuchten Geschwindigkeit abhängig. Für die verletzte Beschränkung wird die Gleichung beispielsweise mit dem Newton-Verfahren gelöst und ein kleineres Intervall bestimmt. Mit dem neuen Intervall werden erneut die restlichen Ruckbeschränkungen überprüft und, falls weiterhin Nebenbedingungen verletzt bleiben, die beschriebenen Schritte wiederholt.

Nachdem keine Ruckbeschränkung mehr verletzt ist, wird mit dem bereits eingeschränkten Intervall dieselbe Herangehensweise auch bei den Beschleunigungsbeschränkungen angewendet.

Mit dieser Methode muss in vielen Fällen das Intervall der Geschwindigkeit v_{i_+} nur zweimal neu berechnet werden: einmal für die obere und einmal für die untere Schranke. Letztendlich liefert das Verfahren ein Intervall für die Geschwindigkeit $v_{i_+} \in [v_{i_+}, \bar{v}_{i_+}]$, durch das keine Nebenbedingung verletzt wird.

3.3 Verbindung von Vorwärts- und Rückwärtzweig

Treffen der Vorwärts- und der Rückwärtzweig nach dem Programmablaufplan in Abb. 3 aufeinander, muss die Übergangsstelle oft noch einmal nachberechnet werden. Durch die Auswahl des Zweiges mit der jeweils geringeren Geschwindigkeit während der Fwd/Bwd-Berechnung, wird eine Verletzung der Beschleunigung beim Übergang vom Vorwärts- zum Rückwärtzweig umgangen. Jedoch tritt hierbei häufig eine starke Ruckverletzung auf. Um diese Verletzung aufzulösen, wird das Geschwindigkeitsprofil lokal um diese Stelle abgesenkt. Dabei wird bei dem Zweig

mit der größeren Geschwindigkeit vorgetäuscht, dass am Übergangspunkt die Berechnung des Intervalls nicht erfolgreich war, was zu einem Absenken des Zweiges führt. Ist dieser Zweig dann erneut bis zur Übergangsstelle berechnet, wird der Ruck überprüft und ggf. wiederum ein Zweig abgesenkt.

Tatsächlich ist die Verbindung der beiden Zweige häufig aufwändiger als die Berechnung der Zweige selbst und lässt sich durch Sonderbetrachtungen stark beschleunigen. Wenn einer der beiden Zweige sehr geringe Beschleunigungen in den letzten Punkten vor der Übergangsstelle aufweist, z.B. weil dort eine Achs- oder Bahngeschwindigkeitsbegrenzung wirkt, ist es sinnvoll, den anderen Zweig über die Übergangsstelle hinauslaufen zu lassen und bereits berechnete Punkte zu überschreiben. Wird der Zweig dann ebenfalls durch die selbe Achs- oder Bahngeschwindigkeitsbegrenzung begrenzt, tritt keine Ruckverletzung an der verschobenen Übergangsstelle mehr auf.

Da das Absenken eines Zweiges sehr viele Iterationen beinhalten kann, ist es teilweise sinnvoll, vorher abzuschätzen, wie viele Stützstellen zurückgegangen werden muss. Diese Abschätzung lässt sich aus der Höhe der Ruckverletzung an der Übergangsstelle ableiten.

Letztendlich kann man hierbei auch beachten, dass eine strikte Einhaltung der Achsruckbeschränkungen nicht immer unbedingt notwendig ist, da die Beschränkungen in der Praxis häufig nur grobe Schätzungen sind.

3.4 Anwendung am Beispiel Kreis

Der vorgestellte Algorithmus wird zunächst auf das in Abschnitt 2.4 vorgestellten Beispiel angewandt. Dabei zeigt sich, dass sich das Geschwindigkeitsprofil ohne Betrachtung von Sonderfällen einfach lösen lässt. Abb. 4 zeigt die Ergebnisse des Beispiels. Es ist das Geschwindigkeitsprofil (rot), die Bahnbeschleunigung (blau) und der Bahnrick (grün) sowohl über der Zeit t (links unten) als auch über dem Bahnparameter s (rechts oben) aufgetragen. Weiterhin lässt sich aus der Abbildung noch die Bahn in der x -

y -Ebene (links oben), die Geschwindigkeit, die Beschleunigung und der Ruck (rechts unten) der beiden Linearachsen X und Y ablesen. Die Bahn ist in vier Berechnungsabschnitte unterteilt. Die lokalen Minima der oberen Grenze der Bahngeschwindigkeit (schwarze Kurve im Diagramm links oben) treten auf dem Kreis bei $s/R = \{\pi/2, \pi, 3\pi/2\}$ auf, wo nur eine Achse die gesamte Bahngeschwindigkeit übernimmt. Die Beschränkungen der Achsen können den horizontalen Linien in den drei Diagrammen (rechts unten) entnommen werden. Die maximale Achsgeschwindigkeit und -beschleunigung wird in den Diagrammen nicht überschritten. Die geringe Verletzung im Achsruck tritt vor allem an den Übergangsstellen zwischen Vorwärts- und Rückwärtszweig auf.

4 Ergebnisse einer Fünfachsbe- wegung

Das im vorherigen Abschnitt vorgestellte Verfahren der approximativen Geschwindigkeitsoptimierung wird im Folgenden auf eine Fünfachsbe-
wegung angewendet. Die Achsen der betreffenden Maschine setzen sich aus drei Linearachsen X, Y und Z und zwei Rundachsen A und B zusammen, $A_k \in \{X, Y, Z, A, B\}$. Die Bahn und der Zusammenhang zwischen dem Bahnparameter s und den fünf Maschinenachsen A_k , der in den Gln. (4) für den Kreis geschlossen angegeben werden konnte, ist für dieses Beispiel lediglich als Datentabelle verfügbar. Eine Zeile der Tabelle enthält für einen Bahnparameter s_i die dazugehörigen Positionen und Ableitungen der Achsen $A_k(s_i)$ bis $A_k'''(s_i)$. Die CNC speichert diese Daten während der Abarbeitung des Programms. Auch die Beschränkungen der Maschinenachsen sind daraus bekannt. Abb. 5 zeigt die Ergebnisse der Berechnung in der gleichen Darstellung wie auch schon bei dem Beispiel der Kreisbahn. Dabei ist die Abbildung in drei wesentliche Teile untergliedert.

In dem Diagramm (links oben) ist die Abbildung der Bahn im Raum mit dem Verlauf des Orientierungsvektors dar-

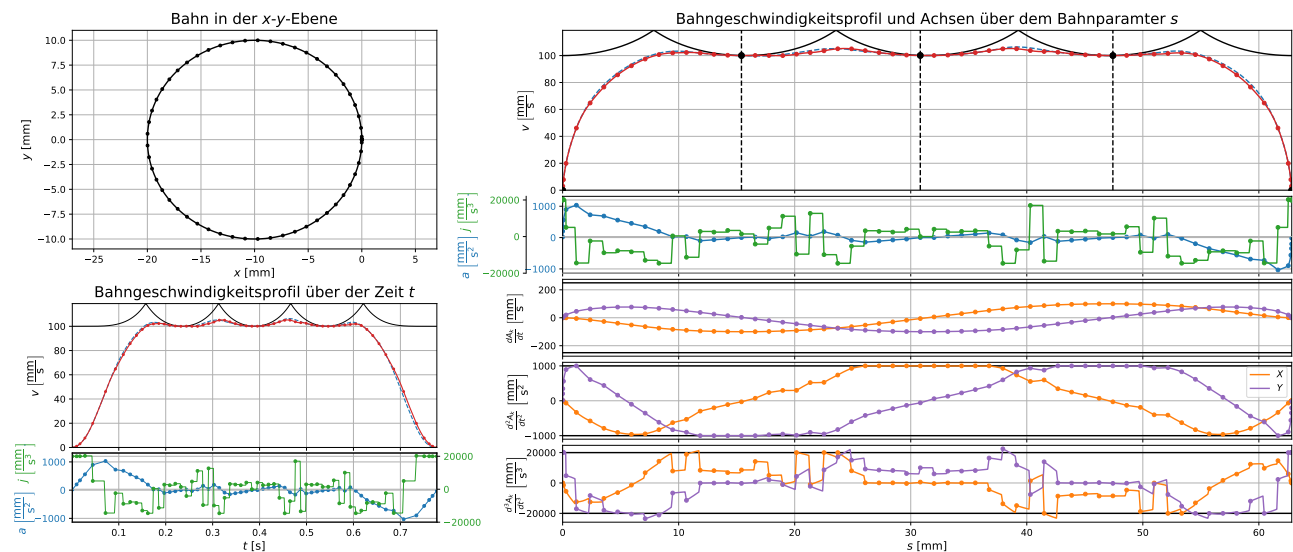


Abbildung 4 Ergebnisse der Geschwindigkeitsoptimierung an dem analytischen Beispiel Kreis in der x - y -Ebene

gestellt. Diesen Orientierungsvektor kann man sich als ein Werkzeug vorstellen, das auf der Bahn entlangfährt. Im ersten Teil der Bewegung $s < 408\text{ mm}$ sind vor allem große Bewegungen erkennbar, wobei es sich um Zustellbewegungen zum Werkstück der Bearbeitung handelt. Das Werkstück wird im zweiten Teil der Bahn bearbeitet, was kleinere Bewegungen mit abrupten Wechsels der Richtung und des Orientierungsvektors zur Folge hat.

In den Diagrammen (rechts) sind sowohl die Bahn- als auch die Achsparameter über dem Bahnparameter s aufgetragen. Im oberen Diagramm zeigt die schwarze Kurve die obere Grenze der Bahngeschwindigkeit nach Gl. (6) aus Abschnitt 3.1. Die vertikalen Markierungen teilen die Bahn in Berechnungsabschnitte und die schwarzen Punkte zeigen die anfänglich berechneten Übergangsgeschwindigkeiten. Die Übergangsgeschwindigkeit bei $s \approx 408\text{ mm}$ resultiert aus einer Unstetigkeit der ersten Ableitung in der Geometrie (Ecke), auf die hier nicht weiter eingegangen wurde. Ausgehend von den Beschränkungen und Unterteilungen wurde das Bahngeschwindigkeitsprofil, dargestellt als rote Kurve, berechnet.

Die blau gestrichene Kurve im oberen Diagramm zeigt die optimale Lösung, also das zeitminimale optimale Geschwindigkeitsprofil. Deren Berechnung ist z.B. in [11] beschrieben. Um sie mit den Ergebnissen des Beitrags vergleichen zu können, ist das Geschwindigkeitsprofil nicht wie in [11] beschrieben als B-Spline definiert, sondern nach den Gleichungen (2). Für die Berechnung der optimalen Lösung wurde *pyOpt* [10] mit dem nichtlinearen Optimierer *Ipopt* [12] verwendet.

Das zweite Diagramm auf der rechten Seite bildet die

Bahnbeschleunigung (blau) und den Bahnruck (grün) ab. In den unteren drei Diagrammen ist je die Achsgeschwindigkeit, -beschleunigung und der -ruck für die drei Linearachsen X , Y und Z und die beiden Rundachsen A und B aufgetragen. Die schwarzen waagerechten Linien symbolisieren wieder deren Beschränkungen.

Der dritte Abschnitt von Abb. 5 (links unten) zeigt ebenfalls die Bahngeschwindigkeit, -beschleunigung und den -ruck, sowohl die zeitoptimale Lösung, diesmal über der Zeit t aufgetragen.

Anhand der Simulation lässt sich feststellen, dass die Berechnung des Geschwindigkeitsprofils einen deutlich höheren Aufwand zur Folge hat. Einerseits resultiert dies aufgrund der höheren Anzahl an Achsen. Hauptsächlich ist der Aufwand jedoch der deutlich komplexeren Geometrie der Bahn geschuldet. So müssen bedeutend mehr Sonderfälle abgehandelt werden, die bei der Berechnung des Kreises nicht notwendig waren. Einige davon sind in Abschnitt 3.3 beschreiben. Der vorgestellte Algorithmus ist unter Python 3.5 implementiert und wurde auf einem *Intel(R) Core(TM) i7-6820HQ @ 2.70 GHz* berechnet. Die Eckdaten der Berechnung, d. h. die Aufteilung in Berechnungsabschnitte, deren Anzahl an Stützstellen, die benötigen Iterationen/Aufrufe des Algorithmus aus 3.2 pro Abschnitt und die Berechnungszeit sind in Tabelle 1 gegenübergestellt. Die Anzahl der Iterationen bis das erste Mal der Vorwärts- und der Rückwärtszweig aneinandergestoßen sind und eine Verbindung der beiden Zweige überprüft wurde, ist in Klammern () angegeben. Vor allem für $s > 408\text{ mm}$ wird ein erschreckend großer Anteil für die Verbindung der Zweige benötigt.

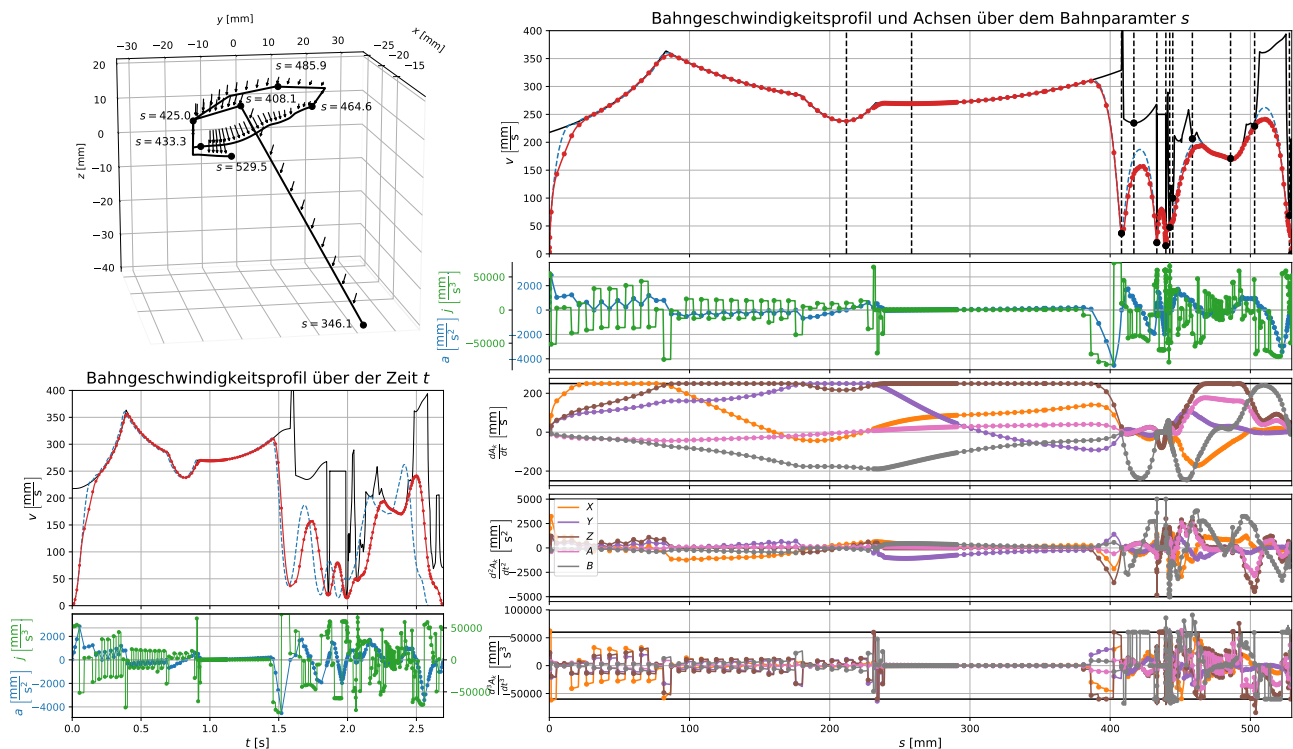


Abbildung 5 Ergebnisse der Geschwindigkeitsoptimierung an dem praktischen Beispiel der 5-Achs-Bewegung im Raum

Tabelle 1 Eckdaten und Zeiten der Berechnung des Geschwindigkeitsprofils aus Abb. 5

Abschnitt	1	2	3	4/5	6	7	8/9/10	11/12/13
Bahnintervall in [mm]	[0, 211.9]	[211.9, 258.4]	[258.4, 408.1]	[408.1, 433.3]	[433.3, 439.7]	[439.7, 442.5]	[442.5, 485.9]	[485.9, 529.5]
Stützstellen	44	25	48	21	32	21	53	48
Iterationen	54 (52)	51 (27)	54 (50)	3429 (76)	7074 (31)	513 (20)	976 (232)	2905 (191)
Berechnungszeit in [s]	0.570	0.679	0.617	60.46	72.22	10.56	16.29	66.45

Aufgrund der Unterteilung in Berechnungsabschnitte ist eine parallele Berechnung auf mehreren Prozessoren einfach möglich. Da in einigen Fällen die Übergangsgeschwindigkeit nicht erreicht wird, müssen dann Abschnitte mehrfach ausgewertet werden.

5 Zusammenfassung

In dem Beitrag wurde eine Erweiterung der schrittweise/iterativen Verfahren für die Berechnung eines näherungsweise zeitoptimalen Geschwindigkeitsprofils vorgestellt. Zum einen unterscheidet sich die Definition der Bahngeschwindigkeit als C^1 Spline mit stückweise konstantem Ruck. Die größten Vorteile liegen in der Parametrisierung nach t für die Abarbeitung in einer CNC und der geschlossenen Lösbarkeit von $a(v_{p+})$ und $j(v_{p+})$ bei einer Parametrisierung nach v_{p+} , vgl. Gln. (3). Für ein numerisches Optimierungsverfahren erweist sich diese Definition jedoch als weniger geeignet, da bei ungeeigneter Wahl von v_{p+} die Größen T , a und j aus den Gln. (3) unbestimmt sein können. Zusätzlich ist die Definition nicht lokal, sondern hängt von allen vorherigen Stützpunkten ab.

Weiterhin wurde eine Methode zur Unterteilung der Bahn in Berechnungsabschnitte vorgestellt. Diese hat bei den vorgestellten Beispielen sehr gute Ergebnisse erzielt. Sie lässt sich sowohl für die schrittweise/iterativen Verfahren, als auch für eine abschnittsweise numerische Optimierung verwenden.

Die geringen Unterschiede des Geschwindigkeitsprofils zwischen dem hier vorgestellten Verfahren und der numerischen Optimierung resultieren voranging aus dem Zusammenfügen der Berechnungszweige. Der deutlich geringere Berechnungsaufwand spricht für die schrittweise/iterativen Verfahren mit den hier vorgestellten Erweiterungen. Trotzdem ist die Berechnungszeit für die Abschnitte vier bis dreizehn in Tabelle 1 deutlich zu hoch. Da auch dies auf das Zusammenfügen der Berechnungszweige zurückzuführen ist, sollte die Verbesserung dieser Methode Gegenstand weiterführender Untersuchungen sein.

6 Literatur

- [1] Beudaert, X.; Pechard, P.; Tournier, C.: *5-Axis tool path smoothing based on drive constraints*. Elsevier, International Journal of Machine Tools and Manufacture. (2011) 51, S. 958–965.
- [2] Beudaert, X.; Lavernhe, S.; Tournier, C.: *Feedrate interpolation with axis jerk constraints on 5-axis NURBS and G1 tool path*. Elsevier, International Journal of Machine Tools and Manufacture. (2012) 57, S. 73–82.
- [3] Bharathi, A.; Dong, J.: *Feedrate optimization for smooth minimum-time trajectory generation with higher order constraints*. Springer, The International Journal of Advanced Manufacturing Technology. (2016) 82, S. 1029–1040.
- [4] Bobrow, J.; Dubowsky, S.; Gibson, J.: *Time-optimal control of robotic manipulators along specified paths*. Sage Publications Sage CA: Thousand Oaks, CA, The international journal of robotics research. (1985) 4, S. 3–17.
- [5] Dong, J.; Stori, J.: *A generalized time-optimal bi-directional scan algorithm for constrained feed-rate optimization*. American Society of Mechanical Engineers, Journal of dynamic systems, measurement, and control. (2006) 128, S. 379–390.
- [6] Dong, J.; Ferreira, P.; Stori, J.: *Feed-rate optimization with jerk constraints for generating minimum-time trajectories*. Elsevier, International Journal of Machine Tools and Manufacture. (2007) 47, S. 1941–1955.
- [7] Herzog, R.; Blanc, P.: *Time-optimal trajectory planning in CNC machining including vibrational behaviour*. IEEE, Control Conference (ECC). (2015), S. 1950–1954.
- [8] Mattmüller, J.; Gisler, D.: *Calculating a near time-optimal jerk-constrained trajectory along a specified smooth path*. Springer, The International Journal of Advanced Manufacturing Technology. (2009) 45, S. 1007.
- [9] Pardo-castellote, G.; Cannon, R.: *Proximate time-optimal algorithm for on-line path parameterization and modification*.
- [10] Perez, R.; Jansen, P.; Martins, J.: *pyOpt: a Python-based object-oriented framework for nonlinear constrained optimization*. Springer, Structural and Multidisciplinary Optimization. (2012) 45, S. 101–118.
- [11] Sencer, B.; Altintas, Y.; Croft, E.: *Feed optimization for five-axis CNC machine tools with drive constraints*. Elsevier, International Journal of Machine Tools and Manufacture. (2008) 48, S. 733–745.
- [12] Wächter, A.; Biegler, L.: *On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming*. Springer, Mathematical programming. (2006) 106, S. 25–57.

Kollisionsvermeidung und Pfadplanung im diskreten Gelenkwinkelraum redundanter Manipulatoren

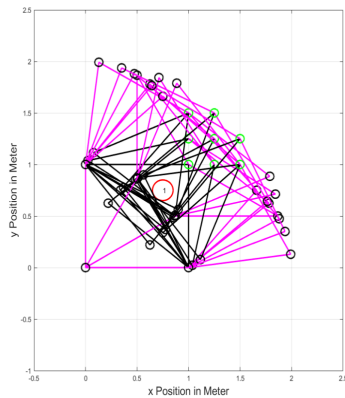
Collision Avoidance and Path Planning in Discrete Joint Space of Redundant Manipulators

Martin Sereinig, Johannes Gerstmayr, Martin Pfurner
Universität Innsbruck, 6020 Innsbruck, Österreich
{martin.sereinig, johannes.gerstmayr, martin.pfurner}@uibk.ac.at

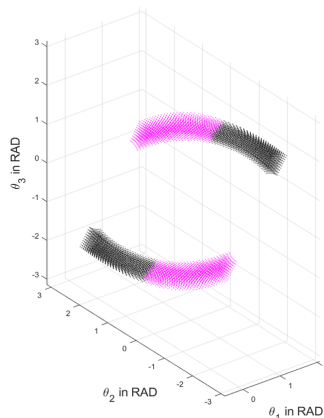
Kurzfassung

Seit einigen Jahren sind redundante Robotersysteme immer mehr in den Fokus von Forscherinnen und Forschern gerückt. Vor allem im Bereich der mobilen Manipulatoren sind redundante Systeme kaum zu vermeiden. Dabei werden Manipulatoren, die oft einen Freiheitsgrad größer sechs besitzen, auf mobile Plattformen mit (Positions-)Freiheitsgrad drei montiert. Das entstehende Gesamtsystem besitzt gewöhnlich einen Freiheitsgrad größer oder gleich neun und muss somit als redundantes System modelliert werden. In [1] wird eine Übersicht über mobile Manipulatoren gegeben. Zusätzlich wird an weiteren hoch redundanten Systemen, wie in [2] beschrieben, geforscht.

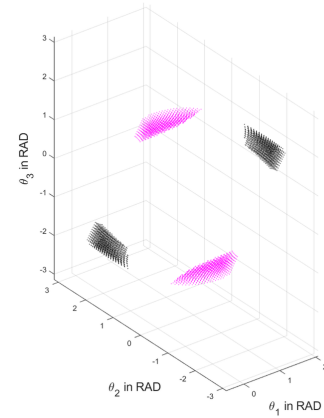
Anders als bei konventionellen, stationären Industrierobotern mit einem Freiheitsgrad gleich sechs, können redundante Roboter, wie oben beschrieben, während der Bewegung der Basis die Position und Orientierung des Endeffektors beibehalten. Eine weitere mögliche Ausnutzung der Redundanzen kann darin liegen, einem Objekt auszuweichen, während der Endeffektor einer bestimmten Bahn folgt. Das Erfüllen von bewegungstechnisch schwierigen Aufgaben, wie das Greifen in einem Rohr, kann durch redundante Systeme möglich gemacht werden.



(a) Mögliche Lösungen zur Erreichung bestimmter Punkte mit $n_{\theta_1} = 4$, $n_x = 3$, $n_y = 3$, $x_{min} = y_{min} = 1$, $x_{max} = y_{max} = 1.5$.



(b) Ausschnitt aus dem Gelenkwinkelraum mit $n_{\theta_1} = 50$, $n_x = 10$, $n_y = 10$, $x_{min} = y_{min} = 1$, $x_{max} = y_{max} = 1.5$ und ohne Objekt.



(c) Ausschnitt aus dem Gelenkwinkelraum mit $n_{\theta_1} = 50$, $n_x = 10$, $n_y = 10$, $x_{min} = y_{min} = 1$, $x_{max} = y_{max} = 1.5$ und mit Objekt.

Bild 1 Untersuchungen der Positionieraufgabe anhand eines planaren Manipulators mit Freiheitsgrad drei.

Das Problem der Bahnplanung und Kollisionsvermeidung durch redundante Konfigurationen kann durch unterschiedlichste Methoden gelöst werden. In [3] beschreiben die Autoren die Lösung basierend auf numerischen Optimierungsalgorithmen unter Nutzung der Pseudoinversen der Jacobi-Matrix. In [4] werden wiederum neuronale Netze genutzt, um eine Lösung des Problems für einen planaren Manipulator mit drei Gelenken zu finden. In [5] und [6] wird ein Ansatz gezeigt, bei dem zur Lösung nicht ausschließlich der kartesische Arbeitsraum des Roboters, sondern auch der Gelenkwinkelraum verwendet wird.

Im Zuge der vorliegenden Arbeit sollen Möglichkeiten der Pfadplanung und Kollisionsvermeidung von redundanten Systemen unter Nutzung des Gelenkwinkelraumes und der inversen Kinematik untersucht und mit bestehenden Lösungen verglichen werden. Erste Schritte werden am Beispiel eines planaren seriellen Manipulators mit Freiheitsgrad drei (wie auch in [4] beschrieben) durchgeführt. Die Orientierung des Endpunktes wird außer Acht gelassen, somit verfügt das System über eine Redundanz vom Grad eins. Bild 1a zeigt mögliche Konfigurationen zum Erreichen der Zielpositionen (grün), wobei der Manipulator um das Objekt (rot) herumgeführt wird. Die durch das Objekt ausgeschlossenen Konfigurationen werden aus dem Gelenkwinkelraum ausgenommen. Bild 1b zeigt einen Ausschnitt aus dem diskreten Gelenk-

winkelraum anhand einer Punktwolke (diskreter Volumskörper) ohne Berücksichtigung eines Objektes. Zum Unterschied wird in Bild 1c dieselbe Punktwolke ohne die durch das Objekt auszuschließenden Konfigurationen der Gelenkwinkel gezeigt.

Ziel ist es, durch eine Kombination von analytischen und numerischen Methoden eine optimale Bahnplanung unter Berücksichtigung von Hindernissen im Gelenkwinkelraum durchzuführen. Im rein numerischen Ansatz entstehen im Gelenkwinkelraum diskrete Hyperquader, welche durch Hindernisse beschnitten werden. Jedes verbleibende, zusammenhängende Netz aus Hyperquadern stellt ein Gebiet dar, welches in der Bahnplanung eingesetzt werden kann. D.h., befinden sich Start- und Endposition innerhalb eines zusammenhängenden Netzes, so ist eine Fahrt möglich. Da dieser Ansatz mit höherer Dimension zu hohem Speicher- und Rechenbedarf führt, sollen die Gebiete möglichst auf Gleichungsebene durch Ungleichungen beschrieben werden. Somit folgt für die Bahnplanung ein zusammenhängender Satz an Gebieten. In dieser Arbeit werden die unterschiedlichen Ansätze, ausgehend von der bestehenden Literatur, getestet und bewertet. Zukünftig sollen analytische Lösungen der inversen Kinematik wie in [7] genutzt werden, um auch höher redundante Systeme im räumlichen Fall untersuchen zu können.

Literatur

- [1] Brandstötter, M.; Mirkovic, D.; Hofbaur, M.: *Mobile Manipulation – Eine altbekannte Technologie findet durch sensitive Robotertechnologie Einzug in die Industrie*. In: C-AR2017 Conference on Automation and Robotics (2017).
- [2] Pieber, M.; Neurauder, R.; Gerstmayr, J.: *An Adaptive Robot for Building in-plane Programmable Structures* IE-EE/RSJ International Conference on Intelligent Robots and Systems (2018), S. 5321–5327.
- [3] Zlajpah, L.; Petric, T.: *Obstacle Avoidance for Redundant Manipulators as Control Problem*. In: Serial and Parallel Robot Manipulators – Kinematics, Dynamics, Control and Optimization (2012), S. 203–230.
- [4] Choong, P. M.; et al.: *Simulation of a Three-links Planar Redundant Manipulator for Obstacle Avoidance Using Neural Networks*. In: IEEE Industrial Electronics and Applications Conference (2016), S. 297–303.
- [5] Bhattacharya, S.; Pivtoraiko, M.: *A Classification of Configuration Spaces of Planar Robot Arms for a Continuous Inverse Kinematics Problem*. In: Acta Applicandae Mathematicae (2015), S. 133–166.
- [6] Stilman, M.: *Global Manipulation Planning in Robot Joint Space With Task Constraints*. In: IEEE Transactions on Robotics (2010), Vol. 26, no. 3, S. 576–584.
- [7] Husty, M.; Pfurner, M.; Schröcker, H.: *A New and Efficient Algorithm for the Inverse Kinematics of a General Serial 6R Manipulator*. In: Mechanism and Machine Theory (2007), S. 66–81.

Agile Montagesystem durch Konsens von Roboter- und Sensornetzwerk – Eine Einführung zur Industrie 4.0

Agile assembly system by consensus of robot and sensor network – An introduction to industry 4.0

Amirreza Shahidi, M. Sc., Dr.-Ing. Dipl.-Wirt.Ing Tim Detert, Prof. Dr.-Ing. M. Hüsing, Prof. Dr.-Ing. Dr. h. c.

Burkhard Corves

RWTH Aachen University, Institut für Getriebetechnik, Maschinendynamik und Robotik, 52072 Aachen, Deutschland,
shahidi@igmr.rwth-aachen.de

Kurzfassung

Agile, frei-vernetzte Montagesysteme, die sich durch die sensorgestützte Kooperation mehrerer mobilen und stationären Roboter auszeichnen, können durch die Vereinigung von Robotik- und Sensornetzwerk als Teil des Internet of Production (IoP) betrachtet werden, welches ein übergreifendes Informations-Netzwerk darstellt. Konkret können diese Montagesysteme durch die Abbildung in einem Meta-Modell befähigt werden als Teil des IoP zu agieren. Dies ermöglicht die Generierung eines sogenannten „Digital-Shadow“ im IoP [1-5].

Am Robotik-Labor des „Institut für Getriebetechnik, Maschinendynamik und Robotik IGMR“ der RWTH Aachen ist ein Montageszenario abgebildet worden, dass sich durch Kooperation zwischen mehrere mobilen und stationären Roboter- und Sensorsystemen darstellen lässt (siehe Abb. 1). Im umgesetzten Montageszenario kooperieren drei Roboter: Ein mobiler Manipulator als Agent, eine mobile Plattform als Logistik-Einheit und ein Portalroboter als stationärer Arbeitsplatz. Die örtliche Referenzierung zwischen diesen Agenten wurde durch ein indoor GPS System (iGPS) realisiert. Die Kommunikation erfolgt über ein gemeinsames Informations-Netzwerk. Über das Informations-Netzwerk können aus anderen Domänen des IoP Erkenntnisse übertragen werden, um einen Vorteil für die Montage zu erzielen.

Im Zuge des Beitrags wird erläutert, wie modulare und flexible Montagesysteme mit reduzierten und allgemeinen Modellen beschrieben werden können, um ihre Verwendung im Internet of Production zu ermöglichen. Die technischen Fähigkeiten werden von Montagesystem basierend auf bekannten Leistungsindikatoren modelliert, um diese Informationen für eine flexible Produktionsplanung zu nutzen.

Als Weiterentwicklung und Erweiterung der Anwendungen des obengenannten Metamodells, ist die Integration von „Augmented Reality AR“ und „Virtual Reality VR“ vorgesehen, um die Verbindung zu weiteren Forschungsfelder, wie z. B. der Fabrikplanung, zu ermöglichen.

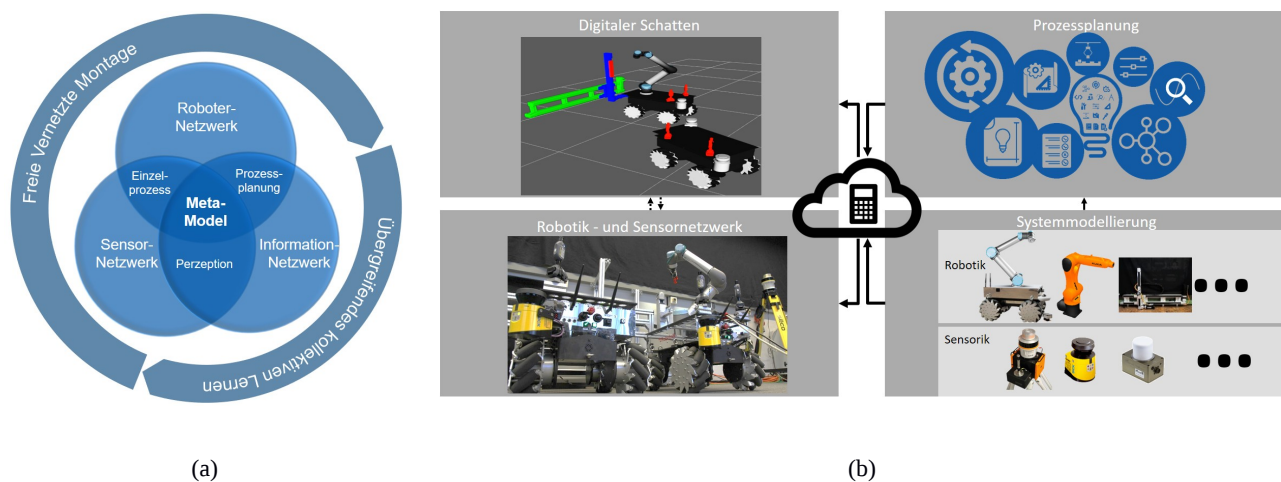


Abb. 1: (a) Vernetzung der Robotik-, Sensorik- und Informationnetzwerke (b) Kommunikationssystem mit mehreren kooperierenden mobilen und stationären Roboter- und Sensorsystemen in der (edge-)Cloud Struktur

Literatur

- [1] Michael Rüßmann; Markus Lorenz; Philipp Gerbert; Manuela Waldner; Jan Justus; Pascal Engel; and Michael Harnisch: Industry 4.0. The Future of Productivity and Growth in Manufacturing. Edited by The Boston Consulting Group.
- [2] Kamarul Bahrin, Mohd Aiman; Othman, Mohd Fauzi; Nor Azli, Nor Hayati; Talib, Muhamad Farihin (2016): Industry 4.0: A Review on Industrial Automation and Robotic. In Jurnal Teknologi 78 (6-13). DOI: 10.11113/jt.v78.9285.
- [3] Lee, Jay; Bagheri, Behrad; Kao, Hung-An (2015): A Cyber-Physical Systems architecture for Industry 4.0-based manufacturing systems. In Manufacturing Letters 3, pp. 18–23. DOI: 10.1016/j.mfglet.2014.12.001.
- [4] Lu, Yang (2017): Industry 4.0: A survey on technologies, applications and open research issues. In Journal of Industrial Information Integration 6, pp. 1–10. DOI: 10.1016/j.jii.2017.04.005.
- [5] Brecher, Christian; Özdemir, Denis (Eds.) (2017): Integrative Production Technology. Theory and Applications. Cham, s.l.: Springer International Publishing. Available online at <http://dx.doi.org/10.1007/978-3-319-47452-6>.

Vergleich von haptischen Steuerungsvarianten zur manuellen Bedienung von Robotersystemen

Comparison of haptic control mechanisms for manual manipulation of robotic systems

Philipp Wabnitz, M.Sc., Dr.-Ing. Andreas Heine, Univ.-Prof. Dr.-Ing. Maik Berger
philipp.wabnitz@mb.tu-chemnitz.de, andreas.heine@mb.tu-chemnitz.de, maik.berger@mb.tu-chemnitz.de
Technische Universität Chemnitz, Professur Montage- und Handhabungstechnik, 09107 Chemnitz, Deutschland

Kurzfassung

Die vorliegende Arbeit beschäftigt sich mit der manuellen Bedienung von Robotersystemen bei der Teach-In-Programmierung und deren Weiterentwicklung hinsichtlich intuitiver und haptischer Steuerungskonzepte. Hierfür werden verschiedene Geräte zur Steuerung eines kleinen Deltaroboters verwendet und bezüglich ihrer Bedieneigenschaften untersucht. Des Weiteren wird aufgezeigt, wie ein haptisches Vibrationsfeedback mittels dieser Bedienmethoden implementiert werden kann. Ziel dieser ersten Variantenstudie ist die Bewertung der Grundkonzepte dieser Geräte in Bezug auf Ergonomie, Systemgrenzen und Fähigkeit zur Ansteuerung von Raumkoordinaten für die Teach-In-Programmierung. Die Ergebnisse stellen die Grundlage für weitere Untersuchungen dar, welche zeigen sollen, inwieweit eine Implementierung der ausgewählten Steuerungskonzepte für einen handelsüblichen Roboter möglich ist und welche Vorteile daraus für kollaborierende und hybride Robotersysteme entstehen.

Abstract

This thesis deals with the manual operation of robot systems during teach-in programming and their further development with regard to intuitive and haptic control concepts. Various devices are used to control a small Delta robot and their operating characteristics are investigated. Furthermore, it is shown how a haptic vibration feedback can be implemented by means of these operating methods. The aim of this first variant study is to evaluate the basic concepts of these devices with regard to ergonomics, system boundaries and the ability to control spatial coordinates for teach-in programming. The results are the basis for further investigations, which should show to what extent an implementation of the selected control concepts for a commercial robot is possible and which advantages arise from it for collaborating and hybrid robot systems.

1 Einleitung

Aufgrund wirtschaftlicher Bedürfnisse und Vorteile hinsichtlich Flexibilität und Zeitersparnis haben sich Robotersysteme in vielen Bereichen der Montage- und Handhabungstechnik bewährt. Für diverse Einsatzszenarien gibt es eine Vielzahl an Modellen und dafür wiederum einen enormen Umfang an Greifersystemen und Peripherie. Damit wird ein hoher Grad an Automatisierung von Arbeitsabläufen erreicht.

Bei der Programmierung dieser Systeme sind jedoch nach wie vor programmiertechnische Kenntnisse und Erfahrungswissen vonnöten. Deshalb werden an Robotersysteme weiterhin Anforderungen an eine leichte Anwendbarkeit und eine kurze Einarbeitungszeit für Programmierer gestellt, welche im täglichen Einsatz Fehler vermeiden sowie Zeit und somit Kosten einsparen.

Zum einen wird es durch die Entwicklung heutiger Roboterarbeitsplätze hin zur kollaborierenden Robotik und dem Einsatz neuartiger hybrider Systeme notwendig, neue Steuerungskonzepte zu evaluieren, welche für diese Szenarien Vorteile bringen [1]–[3]. Zum anderen ist bei jeder Mensch-Maschine-Interaktion die Ergonomie von großer Bedeutung, was sich insbesondere in der kollaborierenden

Robotik zeigt [4]. Die Haptik des Robotersystems spielt hierbei eine zentrale Rolle, denn neben dem subjektiven Befinden bei der Benutzung des Systems können haptische Sinneseindrücke zur Vermittlung von Informationen genutzt und damit die Steuerung eines komplexen Robotersystems vereinfacht werden. In diesem Bereich haben sich sogenannte Feedbacksysteme bereits bewährt, welche es erlauben haptische Informationen nicht nur wahrzunehmen, sondern auch an das System zurückzugeben.

Bei der manuellen Bedienung und insbesondere bei der Teach-In-Programmierung von Robotersystemen ist es notwendig den Tool-Center-Point (TCP) in Absolutwerten vorzugeben oder inkrementell zu verändern, um bestimmte Posen anzusteuern. Hierbei werden Handbediengeräte eingesetzt, welche zur Einstellung von Parametern die Eingabe über einen Touchscreen oder mechanische Eingabeeinheiten wie Schalter oder Drehräder unterstützen. Um eine intuitive und ergonomische Handhabung zu ermöglichen, können in Kombination mit solchen Roboterbediengeräten weitere Steuerungssysteme eingesetzt werden wie beispielsweise sogenannte 6D-Mouse-Eingabeelemente. [5]

2 Stand der Technik

Die Programmierung eines Roboters wird nach Weber in Offline- und Online-Programmierung unterteilt. Während die Offline-Programmierung eine Programmierungsumgebung mit Simulationstools benötigt, geschieht die Online-Programmierung direkt am Roboter selbst. Eine Variante der Online-Programmierung stellt die sogenannte Teach-In-Programmierung dar. Diese Art der Steuerung von Robotersystemen erfolgt vielfach mit sogenannten Programmierhandgeräten (Teach-Panels), welche es erlauben mit einem gewissen räumlichen Abstand Positionierbefehle zu geben und unter Sichtkontrolle ausführen zu lassen. Diese Panels werden in verschiedenen Varianten eingesetzt – zum Beispiel in Form eines Touchpanels, bei denen Eingaben mittels Touchscreen getätigt werden (siehe **Bild 1**). [5]

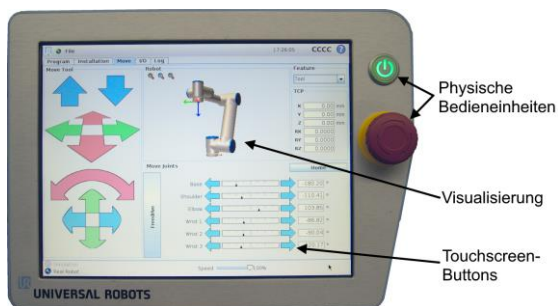


Bild 1 Teach-Pendant des UR5 der Fa. Universal Robots

Die haptische Wahrnehmung des Menschen geschieht mithilfe von Sensorzellen in der Haut und im Bewegungsapparat. Diese Zellen lassen sich nach den Reizen, die sie aufnehmen unterteilen in Merkelzellen, Meissner-, Ruffini- und Vater-Pacini-Körperchen sowie Muskelspindeln. [6] Technisch betrachtet lässt sich die Haptik jedoch auch anders unterteilen und zwar in Kinästhetik und Taktilität. Nach Kern et al. (2014) beschreibt die Kinästhetik die Wahrnehmung der Gelenkpositionen im Raum, der Ausrichtung aller Gliedmaßen, der Orientierung des Körpers und des Muskeltonus. Die Taktilität hingegen fasst Empfindungen von Temperatur, Schmerz und Berührung zusammen, wozu auch die Wahrnehmung von Vibrationen gehört. An der Mensch-Maschine-Schnittstelle werden diese Reize genutzt, um eine bidirektionale Interaktion zwischen dem Nutzer und einem virtuellen Objekt oder einem Teleoperator zu ermöglichen (siehe **Bild 2**). [7] Dieser Beitrag befasst sich vorrangig mit der Vibrotaktilität und dessen Nutzbarmachung, welche in der Literatur bereits umfangreich untersucht wurde [8], [9]. Die Wahrnehmung von taktilen Reizen während der Interaktion führt zu intuitiveren Benutzereingaben, da somit beispielsweise die Einhaltung von Parametergrenzwerten nicht nur visuell oder auditiv, sondern auch haptisch wahrgenommen werden kann. [10]–[12]

Im Hinblick auf neue Bedienkonzepte sind aber auch kinästhetische Eingabemöglichkeiten von Interesse, da sie das Erlernen der Steuerung durch den Einsatz von Gesten auf bekannte Bewegungen reduzieren und somit vereinfachen.

[13]–[15] Versuche haptische Feedbacksysteme bei der Steuerung von Robotersystemen und Greifern einzusetzen sind bereits vielfach in der Literatur beschrieben. [17]–[20]



Bild 2 Bidirektionale Interaktion bei haptischen Interfaces nach Kern et al. und Srinivasan/Basdogan [7], [16]

3 Robotersystem

Aufgrund der genannten Forschungsgegenstände musste ein Roboter gewählt werden, welcher die Anforderungen an einen Einsatz in einem hybriden und kollaborierenden Robotersystem erfüllt. Da in Folgeuntersuchungen der Schwerpunkt auf der Verwendung in einer Hybridstruktur liegt und der Roboter dabei an einem UR5 der Fa. Universal Robots befestigt wird, fiel die Wahl auf eine Delta-kinematik. Hierbei erlauben die drei Achsen eine Positionierung des TCP im Raum jedoch keine Änderung der Orientierung.

Das kostengünstige, leichte und trotzdem den Belastungen gerechte Design der konzipierten Deltakinematik wird durch eine parametrische Konstruktion mit entsprechend wenigen Einzelteilen erreicht (siehe **Bild 3**). Die meisten Bauteile sind mittels FDM (Fused Deposition Modeling) gedruckt und alle weiteren Kauf- und DIN-Teile sind dem Modellbau-Bereich entnommen. Dieser kompakte Delta-roboter setzt sich aus einer zweiteiligen Basis, den Servoarmen, den Koppelstangen und der Endeffektoraufnahme zusammen und erfüllt aufgrund seines Gewichts von 700 g und der hohen Dynamik aufgrund der eingesetzten Servomotoren die an ihn gestellten Anforderungen.

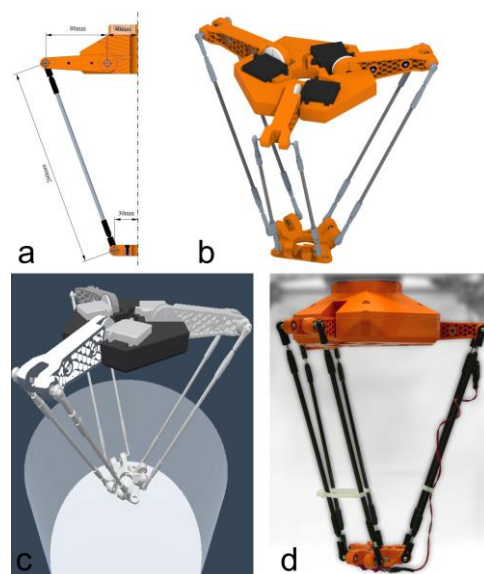


Bild 3 Konstruktion des Deltaroboters. (a) Abmessungen, (b) Rendering, (c) virtuelles Modell und (d) reales Modell

Bild 4 zeigt das Systemschema der Mensch-Roboter-Interaktion. Das eingesetzte Bediengerät gibt eine neue TCP-Position vor, welche in einem handelsüblichen Computer über die inverse Kinematik des Deltaroboters in eine Pose umgerechnet wird. Die dabei ermittelten Winkel werden daraufhin durch die Graupner DES707BBMG Servomotoren angefahren, wobei die Positionsregelung in diesen integriert ist. Ein am Endeffektor befestigter DF9-40-5kg Drucksensor misst die Belastung bei Kontakt mit dem Untergrund oder einem Objekt und diese wird anschließend in Form von Vibrationen an den Nutzer zurückgegeben.

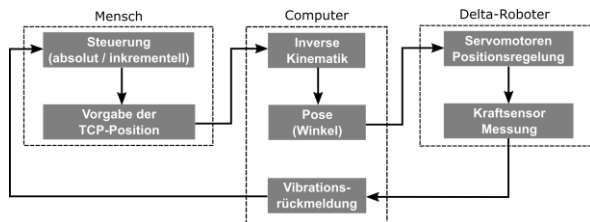


Bild 4 Systemschema für die TCP-Steuerung mit Vibrationsfeedback

Zur Steuerung kommen Komponenten des Elektronikbaukastensystems der Fa. Tinkerforge zum Einsatz (siehe **Bild 5**). Die Kommunikation mit dem Computer übernimmt dabei ein Masterbrick, welcher ebenfalls als Schnittstelle zu einem Analog-In-Bricklet dient, mit dem der Spannungsabfall über dem Drucksensor gemessen wird. Zur Steuerung der Servomotoren wird ein Servobrick verwendet, welches die Positionsvorgaben mittels Pulsweitenmodulation (PWM) an die angeschlossenen Servomotoren weitergibt.

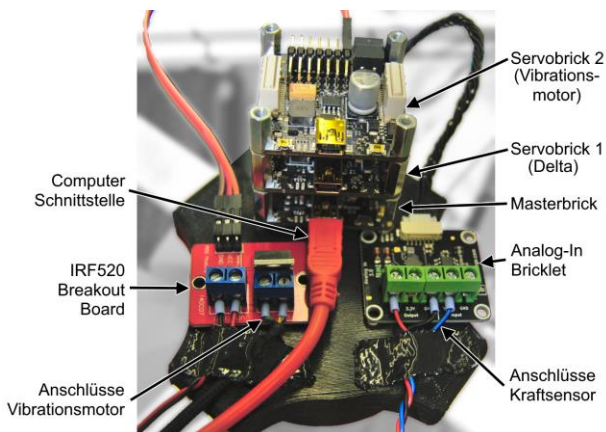


Bild 5 Steuerungskomponenten der Fa. Tinkerforge

Die Vibrationsrückmeldung erfolgt je nach Steuerungskonzept auf unterschiedliche Weise. Für die Bediengeräte, welche selbst keine eingebauten Vibrationsmotoren enthalten, wird ein Vibrationsarmband entsprechend dem von Sklar und Sarter (1999) beschriebenen „Tactor“ verwendet (siehe **Bild 6**) [21]. Hierbei kommt ein 3V DC Münzen-Vibrationsmotor zum Einsatz, welcher mit einem IRF520 (Mosfet) Breakout-Board mittels PWM in seiner Vibrationsstärke reguliert wird (siehe **Tabelle 1**). Die be-

nötigte Versorgungsspannung des IRF520 und die PWM-Signale werden aus einem zweiten Servobrick bezogen.

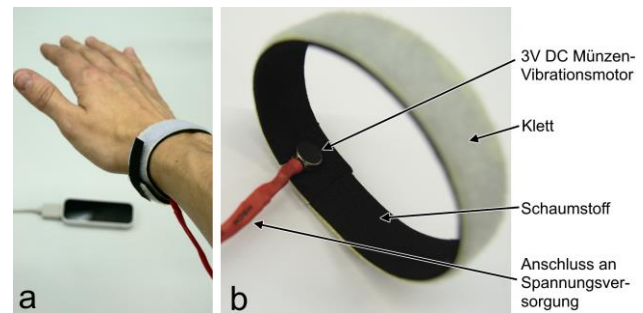


Bild 6 Vibrationsarmband. (a) Einsatz mit optischem Trackingsystem der Fa. Leap Motion und (b) Aufbau

Tabelle 1 Zusammenhang von Drucksensor-Belastung, gemessener Spannung, geregelter PWM und angelegter Betriebsspannung des Vibrationsmotors (Zwischenwerte werden linear interpoliert)

Belastung [N/mm ²]	Gemessene Spannung [mV]	PWM [μs]	Betriebsspannung [mV]
0...0.1	22 ± 3	0	0
0.11	1300	100	40
0.22	3300	10000	4000

4 Steuerungskonzepte

Grundsätzlich lässt sich der Delta-Roboter mit einer simplen Syntax programmieren. Ein Wegpunkt bzw. eine Pose wird mit dem MOVE-Befehl angesteuert. Soll die aktuelle Position mit einem MOVE-Befehl im Programm hinterlegt werden, wird die „Set Point“-Funktion aufgerufen. Für eine Rückkehr zur Nullposition ist eine „Zero“-Funktion implementiert. Der WAIT-Befehl lässt das Roboterprogramm eine bestimmte Zeitspanne in Sekunden warten und ein angeschlossenes Werkzeug kann mittels SET-TOOL-Befehl geschaltet werden.

4.1 Tastatur und Maus

Die händische Eingabe an einem handelsüblichen Computer mit Maus und Tastatur erlaubt eine direkte und inkrementelle Vorgabe des TCP. Mit der Tastatur kann die TCP-Position in Zahlenwerten angegeben oder mit den Pfeiltasten in Schritten angepasst werden. Die Maus stellt ein zusätzliches Hilfsmittel dar, bei dem der TCP in der XY-Ebene positioniert werden kann (siehe **Bild 7**). Das Vibrationsfeedback erfolgt hierbei mit dem Vibrationsarmband.

4.2 Touchscreen

Die Eingabe über den Touchscreen unterscheidet sich nur unwesentlich von der Eingabe mit Maus und Tastatur. Hierbei entfällt lediglich die Schnittstellenperipherie. Ein Betätigen der Buttons mit dem Finger ermöglicht eine intuitive Bedienung, ohne dabei Tasten für bestimmte Aktionen konfigurieren zu müssen. Das Vibrationsfeedback

wird ebenfalls mit dem Vibrationsarmband umgesetzt, jedoch wären auch Touchdisplays möglich, welche bereits Vibrationsmotoren implementiert haben.

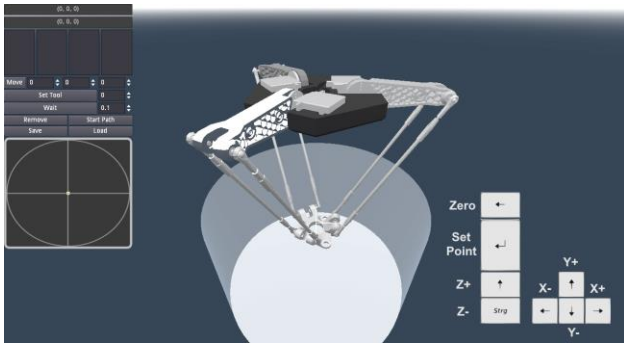


Bild 7 Oberfläche zur Steuerung mittels Maus und Tastatur als auch für die Touchscreenbedienung

4.3 Controller

Die manuelle Eingabe kann wie bei der bereits erwähnten 6D-Mouse mit einem Steuerknüppel (Stick) erfolgen. Dessen Auslenkung wird dann in inkrementelle Schritte in Richtung der dafür definierten Achse des Bezugskoordinatensystems umgesetzt. Um alle drei Achsen des Delta-Roboters zu bedienen, sind mindestens ein Dreiachs-Stick oder zwei Zweiachs-Sticks erforderlich. Deshalb und aufgrund seiner hohen Kompatibilität mit diversen Computersystemen wurde der Xbox360 Controller der Fa. Microsoft ausgewählt. Die Ansteuerung der einzelnen Achsen ist in **Bild 8** zu sehen. Vorteilhaft ist hierbei, dass die Geschwindigkeit, mit der der TCP inkrementell verändert wird, proportional zur Auslenkung gesteigert oder verringert werden kann. Dies erlaubt bei großen Auslenkungen des Sticks eine schnelle Verfahrbewegung und bei kleineren Auslenkungen eine präzisere Ansteuerung von Punkten. Des Weiteren besitzt dieser Controllertyp zwei integrierte Vibrationsmotoren mit unterschiedlichen Schwungmassen, womit verschiedene Vibrationen erzeugt werden können.



Bild 8 Xbox360 Controller der Fa. Microsoft

4.4 Leap Motion

Um eine intuitive Steuerung mittels Handgesten zu ermöglichen, werden vorrangig optische Trackingsysteme eingesetzt. Eines davon, welches speziell für die Erfassung der Handpose entwickelt wurde, ist die sogenannte Leap Motion der Fa. Leap Motion. Diese besteht aus drei

Infrarot-LEDs, dessen Reflexionen mit zwei Infrarot-Kameras erfasst werden. Der Aufbau und die Funktionsweise sind hinreichend bei Weichert et al. (2013) und Marin et al. (2014) beschrieben [22], [23]. Alle gelieferten Hand- und Knöchel-Positionen beziehen sich auf den Mittelpunkt des Leap Motion Controllers, der zwischen den beiden IR-Kameras liegt, direkt über dem zweiten IR-Sender (siehe **Bild 9**). Deshalb wird zur direkten Eingabe des TCP der Ortsvektor des Handmittelpunktes in den Arbeitsraum des Delta-Roboters skaliert übertragen und mit einem Verfahrbefehl als Soll-Position angesteuert. Ein solches System wurde beispielweise bereits von Basily et al. (2014) umgesetzt [24].

Da die Leap Motion auch Informationen über die Fingerposen liefert, können bestimmte Gesten wie bspw. das Schließen und Öffnen der Hand erkannt werden [23]. Diese werden beim Teach-In-Prozess in „Set Point“-Befehle umgesetzt. Das Vibrationsfeedback erfolgt über das Vibrationsarmband.

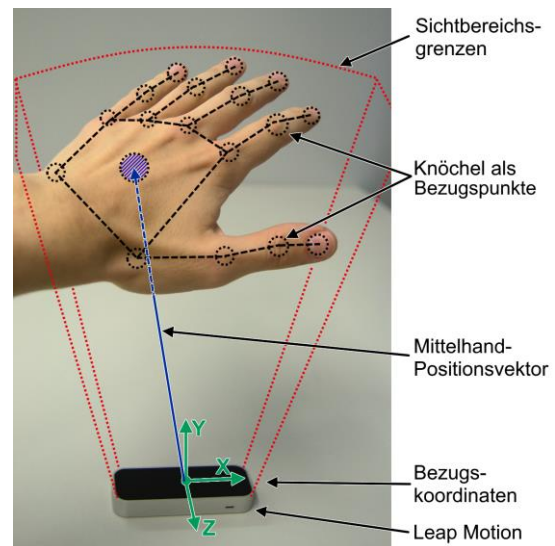


Bild 9 Leap Motion der Fa. Leap Motion (Darstellung nicht maßstabsgerecht)

4.5 Robot Glove

Eine weitere Variante eine Handgestensteuerung zu ermöglichen bietet der Robot Glove der Fa. Lobot. Dieser kostengünstige Handschuh basiert auf einem Arduino kompatiblen Board, welches einen Inertialsensor (IMU), ein Gyroskop, ein Bluetooth-Modul und ein Potentiometer je Finger vereint und somit die Grundvoraussetzungen für die Übertragung der Handpose auf ein Robotersystem erfüllt (siehe **Bild 10**). Die Eingabe von Verfahrbefehlen kann hierbei sowohl über die Beschleunigungswerte der IMU oder durch die Orientierung des Gyroskops erfolgen. Das Vibrationsfeedback wird hierbei ebenfalls über das Armband bereitgestellt, jedoch wäre es möglich der Fingerbewegung mit zusätzlichen Motoren entgegenzuwirken und somit ein kinästhetisches Feedback zu implementieren.

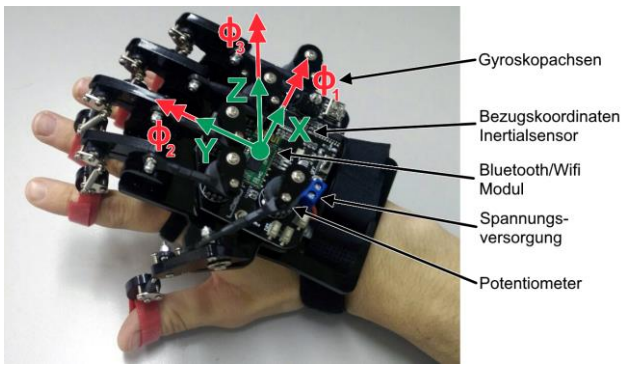


Bild 10 Robot Glove der Fa. Lobot

5 Versuchsdurchführung

Um einen Vergleich der vorgestellten Konzepte zu ermöglichen, wird mit jedem Bediengerät ein vordefinierter Teaching-Prozess durchgeführt (siehe **Bild 11**). Hierbei sollen drei Testpunkte angefahren werden bis ein Kontakt zwischen Werkzeug und Objekt zu Stande kommt, bei dem eine Mindestbelastung von 0.15 N/mm^2 erreicht wird. Da die Andruckkraft entlang der Z-Achse des Delta-roboters ausgeübt werden soll, müssen Zwischenposen angefahren werden, bei denen der TCP über den Testpunkten liegt. Des Weiteren ist auf Kollisionsfreiheit zu achten. Die Versuche werden für jedes Bediengerät zehn Mal durchgeführt, wobei die benötigte Zeit gemessen und anschließend über alle Versuche gemittelt wird, um entstandene Lern- und Trainingseffekte mit einzubeziehen.

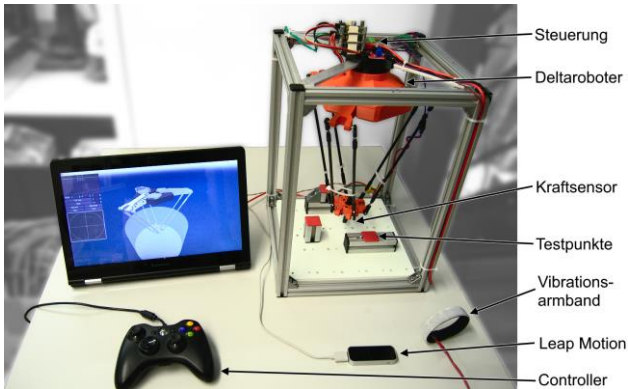


Bild 11 Versuchsaufbau mit einigen Eingabegeräten

Für eine Bewertung des Komforts bzw. die Quantifizierung der dabei auftretenden Beanspruchungen eines Bediensystems können neben objektiven Verfahren wie bspw. EMG, EKG und Gelenkwinkelmessungen auch subjektive Verfahren Verwendung finden. Eine subjektive Beurteilung des Komforts erfolgt am einfachsten durch den Einsatz von Fragebögen oder Interviews mit Probanden, wobei die Borg-Skala ein simples und gleichzeitig vielseitiges Belastungsbewertungsverfahren darstellt. Eine etablierte Variante, die für diese Untersuchung Anwendung findet, ist die für den Gesamtkörper verwendete RPE-Skala, welche aus 15 Anstrengungswerten von 6 („sehr, sehr locker“) bis 20 („sehr, sehr anstrengend“) besteht. [25]

6 Ergebnisse

Die über alle Versuche ermittelten Zeiten sind in **Tabelle 2** zusammen mit den Bewertungen des Komforts aufgelistet. Im Folgenden soll auf einzelne Punkte bei den Untersuchungen eingegangen werden.

Tabelle 2 Vergleich nach benötigter Zeit und subjektiv bewerteter Ergonomie nach Borg (Werte gemittelt über alle Versuche)

	Zeit [mm:ss]	Ergonomie [Borgskala]
Maus und Tastatur	02:20	10
Touchscreen	02:51	9
Xbox360 Controller	01:22	9
Leap Motion	00:48	7
Haptic Hand	-	14

Inkrementelle Eingaben mit der Tastatur können lediglich in der eingestellten Schrittweite getätigt werden. Eine Anpassung der Verfahrensgeschwindigkeit je nach Positionierungssituation ist somit nicht möglich. Dasselbe gilt für die Bedienung mittels Touchscreen, wobei dies durch den Einsatz von Drucksensitivitätsstufen erreicht werden könnte. Des Weiteren war die haptische Rückmeldung eines Tastendrucks beim Tippen auf den Bildschirm nicht gegeben, wodurch Nicht- oder Fehleingaben stattfanden. Hier zeigt sich der Vorteil einer physischen und somit haptisch erfahrbaren Eingabeschnittstelle.

Die Präzision und Wiederholgenauigkeit des Leap Motion Sensors wurden durch Weichert et al. (2013) und Guna et al. (2014) untersucht und bereits kritisiert [22], [26]. Der relativ kleine Erkennungsraum (120°) und auftretende Probleme bei Verschmutzungen sind ebenfalls wahrnehmbare Störfaktoren, die einem alltäglichen Einsatz im Wege stehen. Befindet sich die Hand in einer ungünstigen Lage, z. B. zum Teil hinter einem Gegenstand, so wird deren Position nicht zuverlässig erkannt. Diesen Problemen des optischen Trackings kann mit Systemen aus mehreren Sensoreinheiten entgegengewirkt werden, was bspw. Jin et al. (2016) untersucht haben [27]. Nichtsdestotrotz bietet die Leap Motion eine schnelle und kalibrierungsarme Möglichkeit die Handpose kontaktlos tracken zu lassen.

Ursprünglich sollte der in der Robot Glove verbaute Berschleunigungssensor Verwendung finden. Aufgrund der hohen Messabweichungen und der entstehenden Drift bei Inertialsensoren ist eine Umsetzung in Bewegungsbefehle nicht ausreichend genau möglich. Abhilfe schaffen könnte hierbei ein System, welches auf sensor fusion basiert und optische Messungen mit den Daten der IMU kombiniert. [28]–[30] Die Nutzung des Gyroskops als inkrementelle Steuerungseinheit ist kontraintuitiv und führte zu keiner ausreichend genauen Positionierung des TCP im Testfeld, weswegen keine Zeiten ermittelt werden konnten. Diese Sensordaten könnten stattdessen in weiteren Untersuchungen für die Anpassung der Orientierung bspw. bei einem Hexapoden verwendet werden.

Aufgrund der intuitiven Bedienung und der regelbaren Geschwindigkeit bei unterschiedlichen Positionierungssituationen fallen Zeiten und subjektive Bewertungen des Komforts beim Xbox360 Controller und der Leap Motion deutlich besser aus als bei den anderen Geräten. Diese Steuerungskonzepte haben demnach ein großes Potential die etablierte Touchscreen- oder Tasteneingabe bei der TCP-Positionierung zu ergänzen.

7 Zusammenfassung

Es wurden verschiedene Bediengeräte für die absolute Eingabe und inkrementelle Veränderung des TCP eines Roboters für die Teach-In-Programmierung verglichen. Diese wurden hinsichtlich ihrer Bedieneigenschaften gegenübergestellt und in einem Testszenario im Bezug auf Ausführungsgeschwindigkeit und Komfort untersucht. Es konnte gezeigt werden, dass die haptische Interaktion im Allgemeinen und die vibrotaktile Wahrnehmung im Speziellen durchaus Vorteile bei der Steuerung eines Robotersystems bieten.

Optische Trackingsysteme wie die Leap Motion haben ein großes Potential eine Ergänzung zu einem klassischen Bedienpanel zu werden. Die Positionierung mit der Hand ist intuitiv und schnell. Probleme hinsichtlich Robustheit und Genauigkeit der Gestenerkennung müssten jedoch noch adressiert werden.

Das implementierte Vibrationsfeedback erlaubte bei allen Steuerungsarten eine schnelle Rückmeldung über einen Kontakt mit einem Objekt. Die dabei verwendeten kostengünstigen Drucksensoren bieten für Greifsysteme aufgrund ihrer Größe und der benötigten Auswertelektronik viele Möglichkeiten ein haptisches Feedback zu implementieren.

Weitere Untersuchungen sind nötig, um diese anfänglichen Ergebnisse zu verifizieren. Hierbei stellen sich auch Fragen wie beispielsweise die gemessenen Zeiten mit umfangreicheren Teach-In-Aufgaben skalieren und wie Nutzer die Ergonomie nach einer längeren Anwendung bewerten. Optische Tracker und haptische Handschuhe könnten neue Möglichkeiten der Interaktion mit einem Roboter eröffnen und die Bedienung intuitiver und komfortabler gestalten.

8 Literatur

- [1] B. Matthias, „Die Zukunft der Mensch-Roboter Kollaboration in der industriellen Montage“, S. 13.
- [2] T. Bauernhansl, M. Ten Hompel, und B. Vogel-Heuser, Hrsg., „Mensch-Maschine-Interaktion“, in *Industrie 4.0 in Produktion, Automatisierung und Logistik: Anwendung, Technologien, Migration*, Wiesbaden: Springer Vieweg, 2014, S. 509–523.
- [3] S. Wischmann, „Arbeitssystemgestaltung im Spannungsfeld zwischen Organisation und Mensch-Technik-Interaktion – das Beispiel Robotik“, in *Zukunft der Arbeit in Industrie 4.0*, A. Botthof und E. A. Hartmann, Hrsg. Berlin, Heidelberg: Springer Berlin Heidelberg, 2015, S. 149–160.
- [4] M.-A. Weber und S. Stowasser, „Ergonomische Arbeitsplatzgestaltung unter Einsatz kollaborierender

- Robotersysteme: Eine praxisorientierte Einführung“, *Z. Für Arbeitswissenschaft*, Bd. 72, Nr. 4, S. 229–238, Dez. 2018.
- [5] W. Weber, „Roboterprogrammierung“, in *Industrieroboter: Methoden der Steuerung und Regelung*, 3. Aufl., München: Carl Hanser Verlag GmbH & Co. KG, 2017, S. 106–122.
- [6] M. Grunwald, Hrsg., „Anatomy of receptors“, in *Human haptic perception: basics and applications*, Basel: Birkhäuser, 2008, S. 85–92.
- [7] Hatzfeld und Kern, „Motivation and Application of Haptic Systems“, in *Engineering haptic devices*, New York: Springer, 2014, S. 3–28.
- [8] A. M. Okamura, J. T. Dennerlein, und R. D. Howe, „Vibration feedback models for virtual environments“, in *Proceedings. 1998 IEEE International Conference on Robotics and Automation (Cat. No.98CH36146)*, Leuven, Belgium, 1998, Bd. 1, S. 674–679.
- [9] A. M. Okamura, M. R. Cutkosky, und J. T. Dennerlein, „Reality-based models for vibration feedback in virtual environments“, *IEEE/ASME Trans. Mechatron.*, Bd. 6, Nr. 3, S. 245–252, Sep. 2001.
- [10] M. A. Goodrich und A. C. Schultz, „Human-Robot Interaction: A Survey“, *Found. Trends® Hum.-Comput. Interact.*, Bd. 1, Nr. 3, S. 203–275, 2007.
- [11] J. Tegin und J. Wikander, „Tactile sensing in intelligent robotic manipulation – a review“, *Ind. Robot Int. J.*, Bd. 32, Nr. 1, S. 64–70, Feb. 2005.
- [12] P. Galambos, „Vibrotactile Feedback for Haptics and Telemanipulation: Survey, Concept and Experiment“, *Acta Polytech. Hung.*, Bd. 9, Nr. 1, S. 25, 2012.
- [13] A. Corradini und H.-M. Gross, „Camera-based gesture recognition for robot control“, in *Proceedings of the IEEE-INNS-ENNS International Joint Conference on Neural Networks. IJCNN 2000. Neural Computing: New Challenges and Perspectives for the New Millennium*, Como, Italy, 2000, S. 133–138 Bd.4.
- [14] K. Qian, J. Niu, und H. Yang, „Developing a Gesture Based Remote Human-Robot Interaction System Using Kinect“, *Int. J. Smart Home*, Bd. 7, Nr. 4, S. 6, 2013.
- [15] O. Rogalla, M. Ehrenmann, R. Zollner, R. Becher, und R. Dillmann, „Using gesture and speech control for commanding a robot assistant“, in *Proceedings. 11th IEEE International Workshop on Robot and Human Interactive Communication*, Berlin, Germany, 2002, S. 454–459.
- [16] M. A. Srinivasan und C. Basdogan, „Haptics in virtual environments: Taxonomy, research status, and challenges“, *Comput. Graph.*, Bd. 21, Nr. 4, S. 393–404, Juli 1997.
- [17] J. T. Dennerlein, P. A. Millman, und R. D. Howe, „Vibrotactile Feedback for Industrial Telemanipulators“, S. 7.
- [18] D. S. Pamungkas und K. Ward, „Tele-operation of a robot arm with electro tactile feedback“, in *2013 IEEE/ASME International Conference on Advanced In-*

telligent Mechatronics, Wollongong, NSW, 2013, S. 704–709.

- [19] A. Ajoudani u. a., „Exploring Teleimpedance and Tactile Feedback for Intuitive Control of the Pisa/IIT SoftHand“, *IEEE Trans. Haptics*, Bd. 7, Nr. 2, S. 203–215, Apr. 2014.
- [20] S. Scheggi, F. Morbidi, und D. Prattichizzo, „Human-Robot Formation Control via Visual and Vibrotactile Haptic Feedback“, *IEEE Trans. Haptics*, Bd. 7, Nr. 4, S. 499–511, Okt. 2014.
- [21] A. E. Sklar und N. B. Sarter, „Good Vibrations: Tactile Feedback in Support of Attention Allocation and Human-Automation Coordination in Event-Driven Domains“, *Hum. Factors J. Hum. Factors Ergon. Soc.*, Bd. 41, Nr. 4, S. 543–552, Dez. 1999.
- [22] F. Weichert, D. Bachmann, B. Rudak, und D. Fisseler, „Analysis of the Accuracy and Robustness of the Leap Motion Controller“, *Sensors*, Bd. 13, Nr. 5, S. 6380–6393, Mai 2013.
- [23] G. Marin, F. Dominio, und P. Zanuttigh, „Hand gesture recognition with leap motion and kinect devices“, in *2014 IEEE International Conference on Image Processing (ICIP)*, Paris, France, 2014, S. 1565–1569.
- [24] D. Bassily, C. Georgoulas, J. Güttler, T. Linner, T. Bock, und T. München, „Intuitive and Adaptive Robotic Arm Manipulation using the Leap Motion Controller“, S. 7, 2014.
- [25] G. A. Borg, „Psychophysical bases of perceived exertion“, *Med. Sci. Sports Exerc.*, Bd. 14, Nr. 5, S. 377–381, 1982.
- [26] J. Guna, G. Jakus, M. Pogačnik, S. Tomažič, und J. Sodnik, „An Analysis of the Precision and Reliability of the Leap Motion Sensor and Its Suitability for Static and Dynamic Tracking“, *Sensors*, Bd. 14, Nr. 2, S. 3702–3720, Feb. 2014.
- [27] H. Jin, Q. Chen, Z. Chen, Y. Hu, und J. Zhang, „Multi-LeapMotion sensor based demonstration for robotic refine tabletop object manipulation task“, *CAAI Trans. Intell. Technol.*, Bd. 1, Nr. 1, S. 104–113, Jan. 2016.
- [28] J. Hol, T. Schon, F. Gustafsson, und P. Slycke, „Sensor Fusion for Augmented Reality“, in *2006 9th International Conference on Information Fusion*, Florence, 2006, S. 1–6.
- [29] A. Martinelli, „Vision and IMU Data Fusion: Closed-Form Solutions for Attitude, Speed, Absolute Scale, and Bias Determination“, *IEEE Trans. Robot.*, Bd. 28, Nr. 1, S. 44–60, Feb. 2012.
- [30] E. Arkenbout, J. de Winter, und P. Breedveld, „Robust Hand Motion Tracking through Data Fusion of 5DT Data Glove and Nimble VR Kinect Camera Measurements“, *Sensors*, Bd. 15, Nr. 12, S. 31644–31671, Dez. 2015.

Simulation automatisierter Bauprozesse unter Einsatz von Seilrobotern

Dr.-Ing. Tobias Bruckmann, Universität Duisburg-Essen, Lehrstuhl für Mechatronik, 47057 Duisburg, Deutschland, M.Sc. Arnim Spengler, Universität Duisburg-Essen, Baubetrieb und Baumanagement, 45141 Essen, Deutschland,

Kurzfassung

Mit Beginn der 1970er Jahre führte – ausgehend von Japan – eine erste Welle zu Ansätzen, die Errichtung von Gebäuden auf Basis automatisierter robotischer Systeme umzusetzen [Taylor, 2003]. Höhepunkte waren vollautomatische Fabriken, die als Plattform die Errichtung von Hochhäusern erlaubten [Maeda, 1994]. Diese Systeme wurden bislang jedoch nicht auf andere Gebäudetypen übertragen. Gründe können in den Installationskosten der bis zu 1200t schweren Plattformen und der dazu erforderlichen Infrastruktur vermutet werden.

Hier müssen neue Konzepte gefunden werden, um im Bereich der Errichtung von Gebäuden einen höheren Automatisierungsgrad zu ermöglichen. Getrieben von einer alternden Gesellschaft, dem Fachkräftemangel und dem Mangel an bezahlbarem Wohnraum, wurden in den letzten Jahrzehnten daher Ansätze gesucht, einzelne Bauprozessschritte zu automatisieren [Bock, 2006]. Bis heute hat allerdings auch dies nicht zu einem Wandel in der Baubranche geführt, da die präsentierten Lösungen zumeist nur Insellösungen für einzelne Schritte darstellen und somit ein wirtschaftlicher Einsatz der relativ teuren Geräte nur selten gelang. Bock [2015] schlägt daher vor, auf der Seite der Gebäudeplanung automatisierungsgerechte Gebäudedesigns zu verwenden. Hierin liegt sicherlich ein vielversprechender Ansatz, der nun wiederum der entsprechenden Robotersysteme bedarf, die ein Spektrum an Prozessschritten abdecken können. Diese Roboter müssen kostengünstig, schnell zu errichten und zu demontieren, skalierbar und in Modulen transportierbar sein. Hier bietet sich die Seilrobotertechnologie an.

Seilroboter erlauben es, automatisierte Arbeitsschritte in einem großen Arbeitsraum durchzuführen. Gleichzeitig sind sie sehr leicht und erlauben eine extrem hohe Dynamik. Andererseits erfordert die Verwendung von flexiblen Seilen eine exakte Regelung, die neben einer präzisen Bahnführung auch eine definierte Verspannung in allen Seilen sicherstellt. Um die Frage der grundsätzlichen Einsetzbarkeit von Seilrobotern im Bauwesen zu untersuchen, wurde die Errichtung einer Mauer aus Kalksandsteinen mit Hilfe eines Seilroboters simuliert und experimentell bestätigt. Sowohl die erreichbare Geschwindigkeit als auch die demonstrierte Präzision in der Positionierung von Mauersteinen erscheinen vielversprechend für die praktische Anwendung in der automatisierten Errichtung von Gebäuden [Bruckmann, 2016 & 2018].

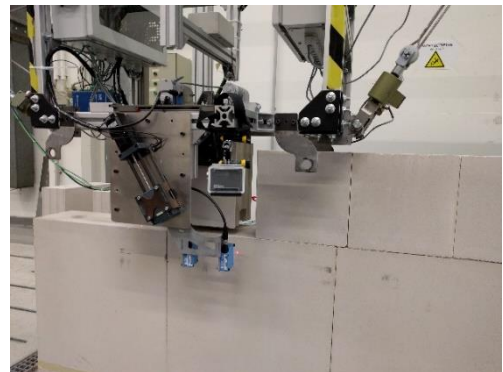


Abb. 1: Seilroboter zum Errichten einer Kalksandsteinmauer

Nach dieser erfolgreichen experimentellen Machbarkeitsstudie sollen in zukünftigen Projekten die Vorteile von Seilrobotern bei der Errichtung von Gebäuden untersucht werden. Dieser Ansatz verspricht Qualitätssteigerungen und Kosteneinsparungen, verlangt aber nach völlig neuen Prozessschritten. Um den Einsatz von Seilrobotern auf der Baustelle daher bereits frühzeitig abschätzen und bewerten zu können, soll eine erste hybride Simulationsumgebung realisiert werden. Diese soll die Simulation eines Seilroboters mit einer Optimierung verbinden, die stets für eine günstige Konfiguration des Seilroboters sorgt. Andererseits soll dies mit einer Baustellenablaufsimulation verbunden werden, die die Baustelle, deren Dauer und Betriebskosten unter verschiedenen Parametern abbildet.

Der Test unterschiedlicher Gebäudetypen wird die technisch-wirtschaftliche Bewertung des Einsatzes von Seilrobotern erlauben. Dabei kann die Simulation zukünftig auch auf andere Robotertypen erweitert werden.

Der Vortrag wird den Stand der Arbeiten sowie die geplanten Projektschritte erläutern und zur Diskussion stellen.

Literatur:

- [Bock, 2006] Bock, Thomas: Construction robotics, Autonomous Robots (2007), Volume 22, Issue 3, pp. 201–209, DOI 10.1007/s10514-006-9008-5, Springer, 2006
- [Bock, 2015] Bock, Thomas: Robot-Oriented Design: The future of construction automation: Technological disruption and the upcoming ubiquity of robotics, Automation in Construction (59), pp. 113–121, 2015
- [Bruckmann, 2016] Bruckmann, Tobias; Reichert, Christopher; Meik, Michael; Lemmen, Patrik; Spengler, Arnim; Mattern, Hannah and König, Markus: Concept Studies of Automated Construction Using Cable-Driven Parallel Robots.

Gosselin, C.; Cardou, P.; Bruckmann, T.; Pott, A. (Hrsg.): Cable-Driven Parallel Robots: Proceedings of the Third International Conference on Cable-Driven Parallel Robots. Cham, Springer International Publishing: 364-375, 2018.

[Bruckmann, 2018] Bruckmann, Tobias; Mattern, Hannah; Spengler, Arnim; Reichert, Christopher; Malkwitz, Alexander and König, Markus: Automated construction of masonry buildings using cable-driven parallel robots. In: Proceedings of 33rd International Symposium on Automation and Robotics in Construction (ISARC 2016), 2016.

[Maeda, 1994] Maeda, Junichiro: Development and Application of the SMART System. In: D.A. Chamberlain (Editor): Automation and Robotics in Construction XI, Elsevier Science B.V., 1994.

[Taylor, 2003] Taylor, Mark Dunlop; Wamuziri, Sam; Smith, Ian: Automated construction in Japan, Civil Engineering 156(1):34-41, DOI: 10.1680/cien.156.1.34.36747, January 2003.

Eine analytische Lösung der inversen Kinematik allgemeiner räumlicher Ketten mit rotativen und zylindrischen Gelenken

Bertold Bongardt, Technische Universität Braunschweig, Institut für Robotik und Prozessinformatik (IRP), 38106 Braunschweig, Deutschland, b.bongardt@tu-bs.de

Kurzfassung

Kürzlich wurde eine neue Methode zur analytische Lösung des inverskinematischen Problems allgemeiner räumlicher Ketten entlang dreier Gelenkachsen beschrieben: für Ketten mit drei Drehgelenken [1], als auch für Ketten mit drei zylindrischen Gelenken [2] ist eine geschlossene Lösung angegeben, die zwei Gelenkwinkelkonfigurationen für eine gegebene, zulässige Endeffektorpose berechnet. Beispielhaft sind in folgender Abbildung zwei Konfigurationen einer Kette mit drei zylindrischen Gelenken gezeigt, die zur gleichen Displazierung des Endeffektors relativ zur Basis führen.

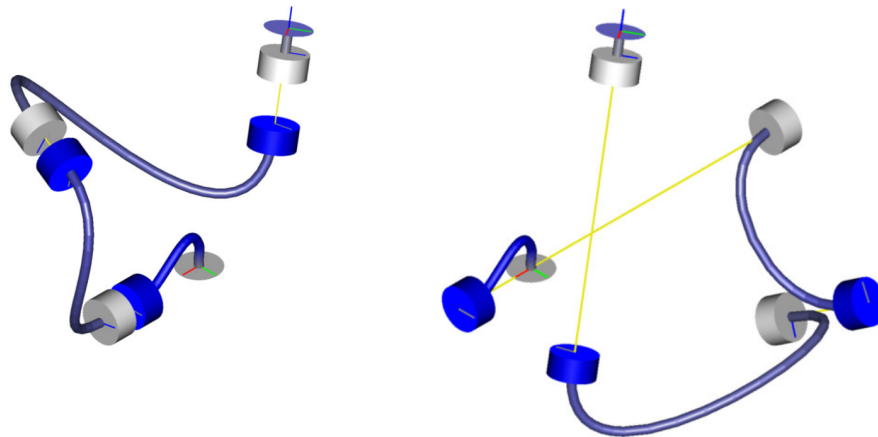


Bild 1 Gelenkkonfigurationen einer räumlichen Kette mit sequentiell-paarweise schiefssymmetrischen Achsen und drei zylindrischen Gelenken, die einen identischen Posenversatz des letzten zum ersten Kettenglied realisieren.

Für den Vortrag im Rahmen der Iftomm-DACH Konferenz ist geplant, die „technischen Werkzeuge“, und ihre „Funktionsweise“, darzustellen, die für die analytischen Lösungen entwickelt worden sind. Diese sind:

- Die Darstellung von Displazierungen über adjungierte (6×6) -Matrizen anstelle homogener (4×4) -Matrizen
- Die zugehörigen trigonometrischen Formen, Verallgemeinerungen der Rotationsformel von Rodrigues, Gauß und Gelman auf Grundlage dreier „Basismatrizen“ bestimmter Funktion
- Die Operation der dualen Normalisierung einer Schraube zu einer normalisierten Geraden in der Form der adjungierten Matrix-Darstellung
- Die geschlossene Lösung der verallgemeinerten trigonometrischen Gleichung $\tilde{a} \cdot \cos \tilde{\phi} + \tilde{b} \cdot \sin \tilde{\phi} = \tilde{c}$, bei der alle Größen und Funktionen „dual“ sind – im Sinne der dualen Zahlen $\tilde{x} = x + \varepsilon \cdot \dot{x}$ mit $\varepsilon^2 = 0$
- Das geometrische Konzept und die praktische Berechnung eines gerichteten, dualen Winkels zwischen zwei räumlichen Geraden relativ zu einer dritten „Referenzgeraden“

Zum Abschluss des Vortrags ist vorgesehen, Pläne für praktische experimentelle Aufbauten zum Testen der Funktionsweise allgemeiner, räumlicher Ketten am Institut für Robotik und Prozessinformatik der TU Braunschweig vorzustellen.

Literatur

- [1] Bongardt, B.: *Novel Plücker Operators and a Dual Rodrigues Formula Applied to the IKP of General 3R Chains*. International Symposium on Advances in Robot Kinematics (2018).
- [2] Bongardt, B.: *The Adjoint Trigonometric Representation of Displacements and a Closed-Form Solution to the IKP of General 3C Chains*. Mechanism and Machine Theory. (Submitted).

Vortragsangebot zur 5. IFToMM D-A-CH Konferenz, 26./27. Februar 2019, Aachen

Autoren: Sebastian Kurz; Prof. Dr.-Ing. Harold Schreiber
Hochschule Koblenz, Fachbereich IW, Konrad-Zuse-Str. 1, 56075 Koblenz
Tel.: +49 – (0)261 – 9528-400
Fax: -499
e-Mail: sekurz@hs-koblenz.de, schreiber@hs-koblenz.de

Stichworte: Biomechanische Kinematik, Prothesentechnik, Handprothetik,
3-D-Druck, Bewegung und Kraftübertragung in Körperteilen

Titel: Kinematische Auslegung von Handprothesen – wie der 3D-Druck die Grenzen der Prothesentechnik neu definiert

Die Produkte moderner Prothesentechnik unterteilen sich in zwei Kategorien:

- täuschend echt aussehende, nicht bewegliche Nachbildungen – kostengünstig, aber im Wesentlichen ohne Funktion,
- myoelektrisch angetriebene Prothesen – funktionsfähig, aber in der Regel für den Großteil der Kunden unerschwinglich.

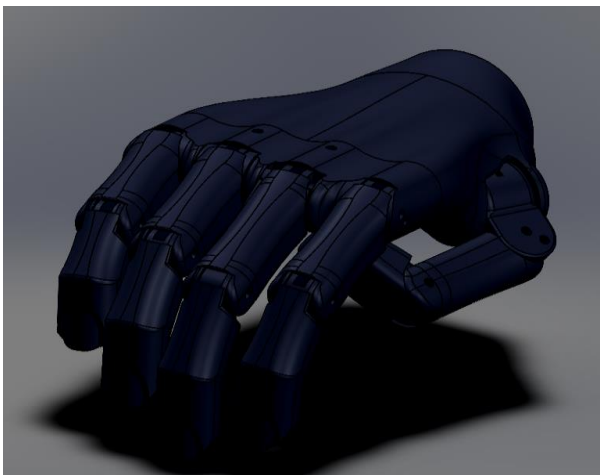
Besonders deutlich zeigt sich dies in der Handprothetik. Kostenbestimmende Faktoren sind dabei insbesondere die komplexen bionischen Geometrien, geringe Stückzahlen und die individuelle Anpassung.

Durch die Technologie der additiven, schichtweisen Materialauftragung ermöglicht der 3-D-Druck eine nahezu völlige Formfreiheit, die mit konventionellen Fertigungsverfahren nicht erreichbar ist. Hierdurch sind Bauteile aus verschiedensten Materialien – Kunststoff, Aluminium, Stahl, Titan, ... – ohne kostenintensive Fertigungsprozesse individuell anpass- und herstellbar.

Im vorliegenden Vortrag wird anhand einer selbst konstruierten Handprothese dargestellt, wie der 3D-Druck insbesondere zur Optimierung von Prothesen eingesetzt werden kann. Der Fokus liegt hierbei auf:

- Antrieb des unteren Fingerglieds,
- Bewegungsübertragung auf die oberen Fingerglieder,
- Rückstellung der Fingerglieder bei Entlastung,
- Schutz des Mechanismus vor Verschmutzung,
- Reduktion des Verletzungsrisikos.

Die Fingerglieder der menschlichen Hand bewegen sich nicht unabhängig voneinander, weshalb bei der prothetischen Nachbildung die kinematische Kopplung der Fingerglieder durch ein geeignetes Koppelgetriebe realisiert werden muss. Der Bauraum dieses Koppelgetriebes ist dadurch extrem begrenzt, dass seine Glieder während der gesamten Bewegung das Fingerinnere nicht verlassen dürfen. Der Antrieb der Beugebewegung der unteren Fingerglieder sitzt hierbei im Unterarm der Prothese und wird über eine bionische, die Muskeln und Sehnen nachbildende Seilkonstruktion übertragen. Eine im Fingergelenk sitzende Blattfeder stellt bei Entlastung die Fingerglieder zurück in die gestreckte Position.



Diese Feder dient zugleich als Abdeckung und Schutz vor Verschmutzung.

Die Problematik der kinematischen Kopplung der Fingerglieder ist in der Prothetik bekannt, jedoch sind die konventionellen Lösungen aufgrund ihrer Struktur auf einen Silikonüberzug angewiesen. Unter anderem um diese Notwendigkeit zu umgehen, lag bei der Konstruktion der hier vorgestellten Handprothese der Schwerpunkt auf der vollständigen Integration der Mechanismen in das Fingerinnere. Auf diese Problematik wird im vorliegenden Vortrag im Detail eingegangen.

Real-Time Substructuring (RTS) zum Testen von Fußprothesen

Christina Insam M.Sc., Technische Universität München, Lehrstuhl für Angewandte Mechanik, christina.insam@tum.de
Prof. Dr. Daniel Rixen, Technische Universität München, Lehrstuhl für Angewandte Mechanik, rixen@tum.de

Kurzfassung

Real-Time Substructuring (RTS) ist eine Methode, um dynamische Systeme zu analysieren. Dabei wird ein Teil des betrachteten mechanischen Systems in einer numerischen Simulation analysiert während der andere Teil des Systems am Prüfstand getestet wird. Um eine Aussage über die Dynamik des Gesamtsystems treffen zu können, müssen beide Simulationen parallel (in Echtzeit), d.h. als Co-Simulation, ablaufen und Informationen miteinander austauschen [2]. Diese Methode eignet sich vor allem für Systeme, bei denen ein Teil sehr schwer zu modellieren ist, da er beispielsweise ein nichtlineares, zeitabhängiges Verhalten aufweist, und der andere Teil für eine Umsetzung am Prüfstand nicht geeignet ist. Da es zur Simulation des gesamten dynamischen Verhaltens sowohl einer numerischen Simulation als auch eines Prüfstands bedarf, spricht man auch von Real-Time Hybrid Substructuring oder Real-Time Hybrid Testing. Diese Methode kommt vor allem im Bereich der Erdbebentechnik verwendet (siehe u.a. [2]), kommt aber auch für die Simulation des Andockens von Satelliten zum Einsatz [1]. Eine neuartige Anwendung des RTS ist das Testen von Fußprothesen. Bei der Verwendung des RTS möchten wir eine Mehrkörpersimulation des Menschen mit der schwer zu modellierenden Fußprothese am Prüfstand in Echtzeit koppeln und das Gesamtsystem Mensch mit Prothese analysieren. So kann bereits in der Entwicklungsphase eine Aussage darüber getroffen werden, wie sich die Prothese in der Interaktion mit dem Mensch verhält. Eine RTS Simulation kann darüber Aufschluss geben, inwieweit der Gang mit der entwickelten Prothese von dem Gang eines gesunden Menschen abweicht.

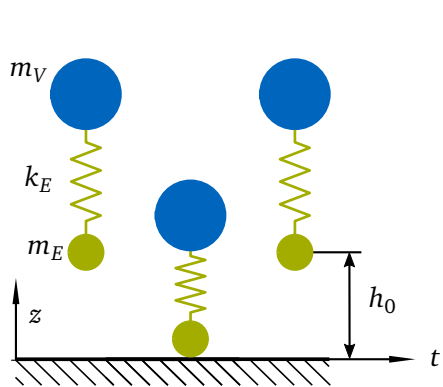
Grundlagen

Beim RTS wird das gesamte mechanische System in zwei Teile aufgeteilt: die virtuelle Komponente, deren Dynamik in einer numerischen Simulation ermittelt wird, und die experimentelle Komponente, die am Prüfstand getestet wird. Da die gesamte Simulation Aufschluss über das dynamische Verhalten des Gesamtsystems geben soll, müssen beide Komponenten in Echtzeit miteinander gekoppelt werden. Die Kopplung findet an der Schnittstelle beider Komponenten statt, wobei Informationen über die Position und Kraft ausgetauscht werden [2]. Der Ablauf der Kopplung ist folgender: Auf das numerische System wirken äußere Lasten sowie Kräfte, die über die Schnittstelle vom experimentellen Teil aufgebracht werden. Unter diesen Lasten wird das virtuelle System für einen Zeitschritt simuliert. Die virtuelle Komponente verformt sich, womit sich auch die Schnittstelle verschiebt. Diese Verschiebung wird auf die experimentelle Komponente durch einen Aktor aufgebracht. Der Aktor kann dann die Kraft messen, die durch die Verschiebung auf die Schnittstelle wirkt. Diese Kraft wird für den nächsten Zeitschritt in der numerischen Simulation als neue Last verwendet. Der Aktor muss demnach positionsgeregelt sein und einen Kraftsensor besitzen. Im Falle, dass der Aktor die gewünschte (in der Simulation berechnete) Position innerhalb eines Zeitschritts (z.B. 1 ms) erreicht und die numerische Simulation die Realität ideal abbildet, ist die Dynamikanalyse mittels RTS perfekt und spiegelt die tatsächliche Dynamik des mechanischen Systems wider. In der Realität weist der Aktor jedoch sowohl Totzeit als auch eine eigene Dynamik auf, wodurch die gewünschte Soll-Position nicht innerhalb eines Zeitschritts erreicht werden kann. Durch diesen Zeitverzug ergeben sich zwei Schwierigkeiten für das RTS: Zum Einen ergeben sich Fehler in der Genauigkeit der Ergebnisse und zum anderen kann die gesamte RTS Simulation instabil werden. Diese Instabilität gilt es zu vermeiden, da dadurch sowohl die experimentelle Komponente als auch die Umgebung beschädigt werden können.

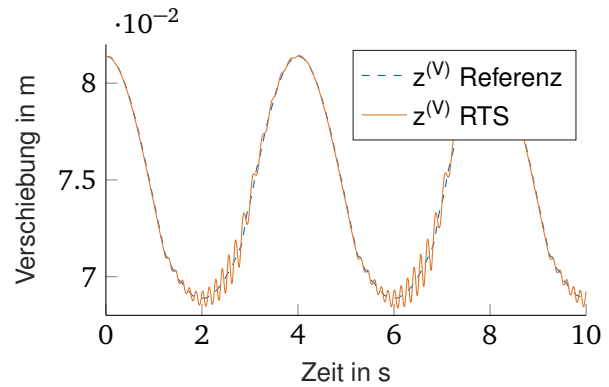
RTS Versuch am einfachen Beispiel

An die Problemstellung des Prothesen-Testens wird mit einem hüpfenden (aus der Höhe h_0) Masse-Feder-Masse-System herangegangen, siehe Abbildung 1a. Die obere Masse m_V wird in der numerischen Simulation (*Simulink*) gelöst und mit dem Feder-Masse-System (k_E , m_E) am Prüfstand gekoppelt. Als Aktor wird ein Hexapod verwendet, der mit 6 Elektromotoren angetrieben und mit *Matlab/dSpace* angesteuert wird. Da aus der Literatur bekannt ist, dass durch die Aktor-Dynamik Instabilität auftreten kann, wird eine Polynom-Extrapolation zweiten Grades als

Kompensation nach [2] angewandt. Der gewählte Lastfall ist Folgender: Die obere Masse m_V soll einer vorgegebenen, sinusförmigen Trajektorie in vertikaler Richtung folgen. Die Trajektorie ist so gewählt, dass sich die untere Masse teilweise im Kontakt befindet und teilweise nicht. Um dieser Trajektorie folgen zu können, wird ein PD-Regler (Parameter K_P und K_D) verwendet. Dieser Ansatz wurde an [3] aus der humanoiden Robotik angelehnt. Je stärker die Parameter K_P und K_D eingestellt werden, desto besser wird der Bahn gefolgt und Störungen haben einen geringeren Einfluss auf die tatsächliche Trajektorie. In den Versuchen ist dies allerdings nicht erwünscht, da ein Mensch beim Gehen nur eine bestimmte maximale Kraft aufbringen kann. Ziel der Versuche ist es, RTS an einem hüpfenden System auszuprobieren und die Anwendbarkeit des RTS für diesen Anwendungsfall zu bestätigen.



(a) Referenzsystem: Masse-Feder-Masse



(b) Ergebnisse des RTS Versuches für eine Trajektorie mit 0.25 Hz

Abbildung 1: Abbildung 1a zeigt das Referenzsystem, das in die virtuelle (blau) und die experimentelle Komponente (grün) aufgeteilt wird. In Abbildung 1b ist die Verschiebung der Schnittstelle der Referenzsimulation (blau, gestrichelt) der realen Verschiebung im RTS Test (rot, durchgezogen) gegenübergestellt. Die Masse m_E befindet sich für $z < 7.5\text{ cm}$ im Kontakt.

Ergebnisse

In Abbildung 1b sind die Ergebnisse der RTS Simulation (rot) der Referenzsimulation des Gesamtsystems (Masse-Feder-Masse, blau) gegenübergestellt. Dem Verlauf der Graphen kann entnommen werden, dass das System instabil wird sobald Kontakt auftritt (Schnittstellen-Verschiebung $z < 7.5\text{ cm}$). Durch die Dynamik der Aktoren und den Impuls, der durch den Kontakt entsteht, schafft es der PD-Regler nicht mehr, der gewünschten Trajektorie zu folgen, wie es in der Referenzsimulation jedoch der Fall ist. Die einfache Polynom-Extrapolation reicht nicht aus, um die RTS Simulation zu stabilisieren und das korrekte Verhalten des Referenzsystems nachzuahmen. Weitere, komplexere Ansätze beinhalten beispielsweise Modellwissen, variable Extrapolations-Horizonte oder andere Regelungsansätze, siehe beispielsweise [1], und müssen nun implementiert und gegebenenfalls erweitert werden. Aufbauend auf diesem eindimensionalen Beispiel können wir dann einfache Modelle des menschlichen Gangs mit einer Prothese am Prüfstand testen.

Literatur

- [1] Boge, T. und Ma, O. „Using advanced industrial robotics for spacecraft rendezvous and docking simulation“. In: *Proceedings - IEEE International Conference on Robotics and Automation* (2011). ISSN: 10504729. DOI: [10.1109/ICRA.2011.5980583](https://doi.org/10.1109/ICRA.2011.5980583).
- [2] Horiuchi, T., Inoue, M., Konno, T. und Namita, Y. „Real-time hybrid experimental system with actuator delay compensation and its application to a piping system with energy absorber“. In: *Earthquake Engineering and Structural Dynamics* 28.10 (1999), S. 1121–1141. ISSN: 00988847. DOI: [10.1002/\(SICI\)1096-9845\(199910\)28:10<1121::AID-EQE858>3.0.CO;2-O](https://doi.org/10.1002/(SICI)1096-9845(199910)28:10<1121::AID-EQE858>3.0.CO;2-O).
- [3] Wittmann, R., Hildebrandt, A. C., Ewald, A. und Buschmann, T. „An estimation model for footstep modifications of biped robots“. In: *IEEE International Conference on Intelligent Robots and Systems* (2014), S. 2572–2578. ISSN: 21530866. DOI: [10.1109/IROS.2014.6942913](https://doi.org/10.1109/IROS.2014.6942913).

Vergleichende Betrachtungen von Bewegungen und Bewegungsmustern zwischen Menschen und humanoiden Robotern

Comparing Observations of Motion and Motion Patterns between Humans and Humanoid Robots

Nora-Sophie Staufenberg, TU München, Lehrstuhl für Angewandte Mechanik, nora-sophie.staufenberg@tum.de

Johanna Vielemeyer, Friedrich-Schiller-Universität Jena, Lehrstuhl für Bewegungswissenschaft und Trainingswissenschaft, johanna.vielemeyer@uni-jena.de

Eric Griessbach, Friedrich-Schiller-Universität Jena, Arbeitsbereich Sportpsychologie, eric.griessbach@uni-jena.de

Dr. Roy Müller, Klinikum Bayreuth GmbH, roy.mueller@uni-jena.de

Prof. dr.ir. Daniel Rixen, TU München, Lehrstuhl für Angewandte Mechanik, rixen@tum.de

Kurzfassung

Wir Menschen gehen scheinbar mühelos auf ebenen Flächen und sogar über unebenes Gelände und sind in der Lage, uns nach kleinen Störungen wieder zu stabilisieren. Für humanoide Roboter stellt Gehen gerade auf unebenem Gelände dagegen immer noch eine Herausforderung dar. Es stellt sich die Frage, was die humanoide Robotik dabei vom Menschen und dessen Bewegung lernen kann, ob also bestimmte menschliche Bewegungsstrategien auf humanoide Roboter übertragbar sind. Um dieser Frage nachzugehen, ist zunächst der Vergleich der Bewegungen und Bewegungsmuster von Menschen und humanoiden Robotern sinnvoll.

Der Gang des Menschen ist eine zyklische Bewegung, man spricht vom sogenannten Gangzyklus. Ein solcher Gangzyklus besteht aus zwei Schritten. Beim menschlichen Gehen fungiert immer ein Bein als Standbein (Standphase) und das andere als Schwungbein (Schwungphase). Während der Standphase befindet sich der Fuß in Kontakt mit dem Boden und es entsteht eine Bodenreaktionskraft. In der Schwungphase wird der Fuß von der aktuellen Standposition zur nächsten bewegt, ohne dabei den Boden zu berühren. [1, 2] Man unterscheidet Gehen und Rennen häufig dadurch, dass es beim Rennen eine Flugphase gibt, in der kein Bein in Kontakt mit dem Boden ist. In diesem Bericht wird ausschließlich Gehen betrachtet, sowohl für den Menschen als auch für den humanoiden Roboter. Beim zweibeinigen Gehen wird die Standphase wiederum unterteilt in eine Einzelstützphase und eine Doppelstützphase, je nachdem, ob sich nur eins oder beide Beine in Kontakt mit dem Boden befindet und belastet ist. [1, 2] Zur Analyse des menschlichen Gangs werden in der Bewegungswissenschaft unter anderem die Bodenreaktionskräfte, der Druckpunktverlauf und die Bewegung des Schwerpunkts herangezogen [2].

In diesem Bericht werden die Bewegungen beim Gehen des Menschen (z. B. [5]) mit der Bewegung eines humanoiden Roboters verglichen. Hierzu wird der humanoide Roboter *Lola* verwendet, welcher am Lehrstuhl für Angewandte Mechanik entwickelt wurde [3, 4].

Um die Bewegung und die zugrundeliegenden Bewegungsmuster zu vergleichen, werden unter anderem die zeitliche Aufteilung der Schritphasen, die Bewegung des Schwerpunkts und die Bodenreaktionskräfte verglichen.

Literatur

- [1] Perry, J.: *Ganganalyse - Norm und Pathologie des Gehens*. München: Urban & Fischer, 2003, ISBN 3437471104
- [2] Whittle, M.: *An Introduction to Gait Analysis*. Edinburgh: Butterworth-Heinemann, 2008, ISBN 9780750688833
- [3] Buschmann, T.: *Simulation and Control of Biped Walking Robots*, Dissertation, Technische Universität München, 2010
- [4] Lohmeier, S.: *Design and Realization of a Humanoid Robot for Fast and Autonomous Bipedal Locomotion*. Dissertation, Technische Universität München, 2010
- [5] Müller, R.; Tschiesche, K.; Blickhan, R.: *Kinetic and kinematic adjustments during perturbed walking across visible and camouflaged drops in ground level*. Journal of Biomechanics 47 (2014), S. 2286 - 2291, doi.org/10.1016/j.jbiomech.2014.04.041

Konzept einer interdisziplinären GUI für die Befundung und Diagnose von Gangstörungen bei der Therapieplanung von Schlaganfallpatienten — das RehaBoard-Projekt

Anna Sondermann¹, Francisco Geu Flores¹, Harald Hefter², Marcus Jäger³, Andrés Kecskeméthy¹, Dominik Raab¹, Mario Siebler⁴, Dörte Zietz⁵, Silke Gegenbauer⁶

¹ Lehrstuhl für Mechanik und Robotik, Universität Duisburg-Essen, Duisburg

² Klinik für Neurologie, Heinrich-Heine-Universität Düsseldorf, Düsseldorf

³ Klinik für Orthopädie und Unfallchirurgie, Universitätsklinikum Essen, Essen, Deutschland

⁴ Neurologie, Fachklinik Rhein/Ruhr für Herz/Kreislauf- und Bewegungssystem GmbH & Co. KG, Essen

⁵ Hochschule für Gesundheit, Bochum, Deutschland

⁶ ITBB GmbH, Neukirchen-Vluyn

Kurzfassung

Gangstörungen von Schlaganfallpatienten stellen eine sehr häufige komplexe Folge mit hohem Leidensdruck der Betroffenen und Angehörigen durch Einschränkung der Teilhabe dar. Aufgrund des soziodemographischen Wandels ist mit wachsenden Fallzahlen sowie mit zunehmenden Komplikationen zu rechnen [1]. Mit ca. 196.000 erstmaligen und 66.000 (Stand 2008) wiederholten Schlaganfällen pro Jahr zählt der Schlaganfall dabei zu den häufigsten Erkrankungen in Deutschland und ist die häufigste Ursache für mittlere und schwere Behinderungen wie u.a. Gangstörungen. Neue Behandlungsmethoden (Arzneimittel, Orthesen, physiotherapeutische Maßnahmen, minimal-invasive und operative Behandlungstechniken) haben in den letzten Jahren fachspezifisch erhebliche Fortschritte sowohl in der akuten als auch in der chronischen Behandlung erzielt. Wissenschaftliche Daten weisen demgegenüber darauf hin, dass Schlaganfallpatienten mit Gangstörungen erheblich von der Anwendung multimodaler und interdisziplinär abgestimmter Therapiekonzepte aufgrund der sich ergebenden Synergismen profitieren könnten [5]. Was bisher daher fehlt, ist eine interdisziplinäre Behandlungsplanung, die Synergieeffekte optimal nutzt.

Im vorliegenden Leitmarkt-Projekt “RehaBoard” arbeiten Experten der Disziplinen Neurologie, Orthetik, Orthopädie und Physiotherapie zusammen mit Ingenieurwissenschaften an einer computergestützten Assistenzumgebung für die interdisziplinäre Diagnose und Behandlungsplanung von Gangstörungen nach Schlaganfall. Bestehende Ansätze basieren auf einer reinen numerischen Auswertung der Bewegungen und haben sich aufgrund der fehlenden Einbeziehung der klinischen Benutzer in die Entwicklung nicht etabliert [2] [3]. Ziel der vorliegenden Arbeit ist es daher, die Sichtweisen der einzelnen Disziplinen in einer einheitlichen interdisziplinärer Gesamtperspektive zu bündeln.

Zu diesem Zweck wurde eine systematische Herangehensweise basierend auf Sharp et al. [4] ausgewählt. In einem ersten Schritt wurden die Bedürfnisse der Anwender analysiert. Hierfür wurden zum einen die Vorgehensweisen bei der Befundung beobachtet. Zum anderen wurden semi-strukturierte Interviews durchgeführt. Die Interviews wurden zur besseren Auswertung als Audiodatei aufgezeichnet und anschließend gemeinsam mit den Daten der Befundungen qualitativ analysiert. Hierzu wurde ein offener Codierungsprozess nach Strauss und Corbin verwendet [4], wodurch die zentralen Kriterien der Disziplinen herausgefiltert wurden. Als Ergebnis zeigt sich, dass die einzelnen Disziplinen unterschiedliche Aspekte beobachten (Tabelle 1).

(A) Physiotherapie	(B) Neurologie	(C) Orthopädie	(D) Orthetik
“Als erstes schaue ich mir das Video an, um einen allgemeinen Eindruck zu kriegen: vom Gang, von der Sicherheit, von Auffälligkeiten. [...] Dann gehe ich in die kinematischen Daten rein.”	“Ich muss mir über die Position, in der der Körper ist, sofort einen Überblick verschaffen und erkennen, was meiner Meinung nach das primäre Problem des Patienten ist.”	“Schmerz kann sehr viel verändern, und wenn man die Information nicht hat, dann kann man die Ganganalyse nicht für die Diagnostik verwenden.”	“Viele Dinge sehe ich dem Patienten direkt an, wenn er vor mir läuft. [...] Welche Störungen er hat [...], man hat einen Blick dafür.”

Tabelle 1 Hauptkriterien der Disziplinen bei Beobachtung des Ganges von Schlaganfallpatienten

Aus der Bedürfnisanalyse wurden vier Aspekte identifiziert, die für eine interdisziplinäre Befundung des Gangbildes nach Schlaganfall relevant sind: (A) die Ganganalysedaten mit Gelenkwinkelverläufen und Schrittparametern, (B) die

Oberkörperbewegung einschließlich Arme, (C) das Schmerzempfinden, sowie (D) das Patientenvideo für das Gangbild. Daneben wurde eine Avatar-Simulation als wesentlich für das Verständnis der inneren Bewegungen im Skelettsystem identifiziert. Diese wurden in einer einheitlichen Visualisierungsoberfläche in einem Prototyp zusammengestellt (Bild 1).

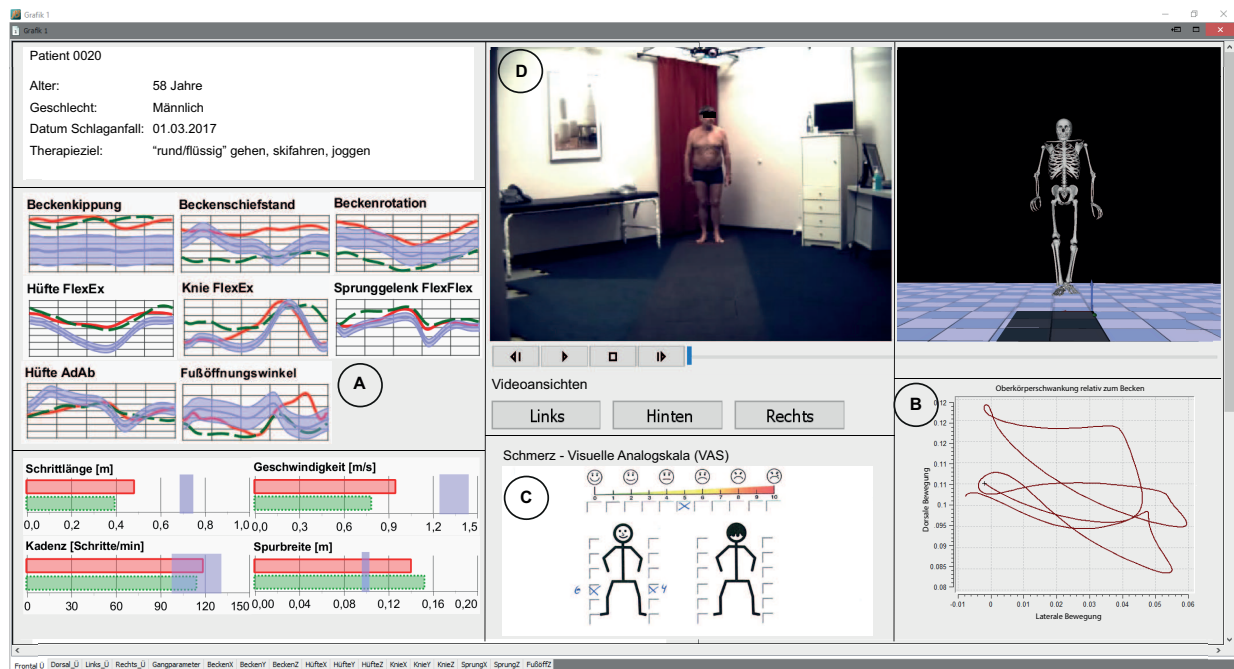


Bild 1 Prototyp der interdisziplinären GUI zur Visualisierung des Gangbildes nach Schlaganfall

Um das Ziel einer benutzerfreundlichen, klinisch einsetzbaren Anwenderoberfläche für die Diagnose und Maßnahmenplanung bei Gangstörungen nach Schlaganfall zu erhalten, wird der Prototyp sowohl in den gemeinsamen Befundungen eingesetzt als auch in einzelnen Interviews weiter verfeinert. In einem iterativen Verfahren soll dieser dabei um die von den Experten zusätzlich genannten Kriterien ergänzt werden, um so praxistauglich zu werden.

Das Projekt „RehaBoard“ wird aus Mitteln des Europäischen Fonds für regionale Entwicklung (EFRE) gefördert.

Literatur

- [1] Foerch, C., Misselwitz, B., Sitzer, M., Steinmetz, H. und Neumann-Haefelin, T.: *Die Schlaganfallzahlen bis zum Jahr 2050*, Deutsches Ärzteblatt International, 105, Seiten 467–473, 2008
- [2] Simon, S. R.: *Quantification of human motion: Gait analysis—benefits and limitations to its application to clinical problems*, Journal of Biomechanics, Volume 37, Issue 12, Seiten 1869-1880, 2004
- [3] van den Bogert A. J., Geijtenbeek T., Even-Zohar O., Steenbrink F. und Hardin E. C.: *A real-time system for bio-mechanical analysis of human movement and muscle functions*, Med Biol Eng Comput 51(10), Seiten 1069–1077, 2013
- [4] Sharp, H.; Rogers Y. und Preece, J.: *Interaction design : Beyond human computer interaction*, John Wiley & Sons, New York, 2009
- [5] Ch. Dohle, J. Quintern, S. Saal, K. M. Stepha, R. Tholen, H. Wittenberg S.: *e-Leitlinie "Rehabilitation der Mobilität nach Schlaganfall (Re-MoS)"*, Neurol Rehabil, 21(4): 179–184, 2015

Smart and Energy-efficient Actuators based on Shape Memory Alloys and Dielectric Elastomers

Intelligente und Energieeffiziente Aktoren basierend auf Formgedächtnislegierungen und Dielektrischen Elastomeren

Dr.-Ing. Paul Motzki, ZeMA – Zentrum für Mechatronik und Automatisierungstechnik gGmbH, Universität des Saarlandes, 66121 Saarbrücken, Deutschland, p.motzki@zema.de

Dr.-Ing. Steffen Hau, Universität des Saarlandes, Lehrstuhl für Intelligente Materialsysteme, 66121 Saarbrücken, Deutschland, steffen.hau@imsl.uni-saarland.de

Prof. Dr.-Ing. Stefan Seelecke, Universität des Saarlandes, Lehrstuhl für Intelligente Materialsysteme, 66121 Saarbrücken, Deutschland, stefan.seelecke@imsl.uni-saarland.de

Kurzfassung

In times of a general widespread demand for more functionality in products and at the same time growing restrictions, engineers are faced with more complex and challenging design tasks. For the development of novel or improved products, more and more engineers start looking into smart materials to replace state-of-the-art actuators or sensors or to design new multifunctional and intelligent actuator-sensor-systems. Smart or active materials like shape memory alloys (SMAs) or electro-active polymers (EAPs) change their properties in response to external fields. These property changes can either be exploited to produce movement and force for actuation or for quantitative measurements like a specific sensor. Aside from their combined actuation and sensing capabilities, smart materials can offer significant savings in weight and construction space because of their high energy and power densities. Actuator systems based on smart materials have the potential to generate very energy-efficient solutions in comparison to electric, electromagnetic, hydraulic or pneumatic actuators and operate without noise and emissions. The materials' specific and flexible form factors enable completely new actuator designs and implementation and current topics of interest such as predictive maintenance and condition monitoring can be addressed by using their sensing abilities. Even further, this "self-sensing" may be utilized in control algorithms for closed-loop position control without the use of additional external sensors.

This work has a specific focus on making actuators based on SMAs and dielectric elastomers (DEs) energy-efficient and demonstrating their properties in exemplary applications. In the first part, a novel bi-stable SMA actuation principle [1-3] is introduced. The actuator is based on SMA wires, which when heated to a specific transformation temperature undergo a phase transformation. The phase transformation causes a contraction of the SMA wire of several percent and the actuator is able to perform mechanical work. SMAs have the highest known energy density of all known actuation principles [4], which leads to very compact and lightweight design solutions. The bi-stability in this actuator leads to significant savings in energy consumptions compared to state of the art SMA actuators. This actuation principle is then exemplary used in the application of a vacuum gripper system. In this system, SMA actuators control and drive suction cups so that no compressed air system is needed [5].

The second part introduces the principle of dielectric elastomer actuators. Electrically, these actuators present a deformable capacitor. Actuation is achieved by loading and unloading this capacitor, which also leads to two quasi energy-free states [6]. Like SMAs, DE actuators (DEAs) show a lot of promise to reduce construction space and weight in applications. Compared to SMA actuators, which are known for high forces but limited actuation frequency (1-10 Hz), DEAs can be activated up to several kHz, but in a lower force region. The potential of DEAs is displayed at the application example of a pneumatic valve. In a first step, a commercial electromagnetic valve is evaluated in a typical work cycle and the pneumatic characteristics and the energy consumption are documented. The electromagnetic drive in the valve is then replaced by a DEA and the same working cycle is used for its characterization. Comparison between the two valves shows identical pneumatic performance, while the DE valve's energy consumption is reduced by a factor of 500 [7].

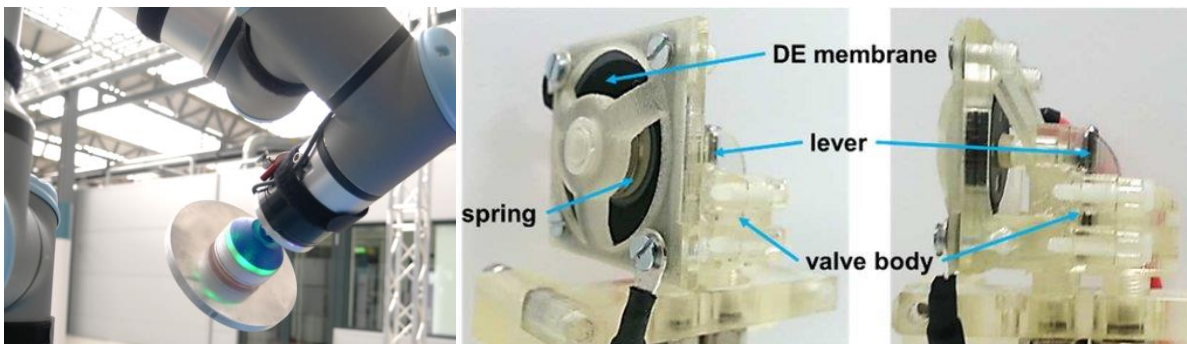


Fig.1: Bi-stable SMA based vacuum suction cup attached to a robot (left) and the DEA driven pneumatic valve (right).

Literatur

- [1] P. Motzki and S. Seelecke, "Bi-stable SMA Actuator," in *Actuator 16 - 15th International Conference on New Actuators*, 2016, pp. 317–320.
- [2] P. Motzki and S. Seelecke, "Bistabile Aktorvorrichtung mit einem Formgedächtniselement," DE 10 2016 108 627 A1, 2016.
- [3] P. Motzki and S. Seelecke, "BISTABLE ACTUATOR DEVICE HAVING A SHAPE MEMORY ELEMENT," WO 2017/194591 A1, 2016.
- [4] H. Janocha, *Unkonventionelle Aktoren - Eine Einführung*, 2. München: Oldenburg Verlag, 2013.
- [5] F. Welsch, S.-M. Kirsch, P. Motzki, M. Schmidt, and S. Seelecke, "Vacuum Gripper System Based on Bistable SMA Actuation," in *ASME 2018 Conference on Smart Materials, Adaptive Structures and Intelligent Systems*, 2018, p. V001T04A014.
- [6] M. Hodgins, A. York, and S. Seelecke, "Experimental comparison of bias elements for out-of-plane DEAP actuator system," *Smart Mater. Struct.*, vol. 22, p. 094016, 2013.
- [7] M. Hill, G. Rizzello, and S. Seelecke, "Development and experimental characterization of a pneumatic valve actuated by a dielectric elastomer membrane," *Smart Mater. Struct.*, vol. 26, no. 8, p. 085023, 2017.

Optimale Synthese nachgiebiger Mechanismen für die Robotik

E. J. Wehrle¹, J. Achleitner², J. Reinisch²

¹ Freie Universität Bozen, Bozen, Südtirol, Italien, Erich.Wehrle@unibz.it

² Lehrstuhl für Luftfahrtsysteme, Technische Universität München, {Johannes.Achleitner, Joseph.Reinisch}@tum.de

5. IFToMM D-A-CH Konferenz 2019

26.–27. Feber 2019

RWTH Aachen

Als Ingenieure streben wir danach, die leichtesten, schnellsten und effizientesten Strukturen und Mechanismen zu entwerfen. Nachgiebige Mechanismen haben eine Reihe von Vorteilen, um dieses Vorhaben zu erreichen, vor allem in der Robotik. In dieser Arbeit kommen Optimierungsmethoden zum Einsatz — vor allem die Topologieoptimierung — für die Synthese optimaler nachgiebiger Mechanismen mit geringem Gewicht. Neuartige additive Herstellungsmethoden, d.h. dreidimensionales Drucken, wie Schmelzschichtungsverfahren und selektives Lasersintern ermöglichen die Realisierung der dabei entstehenden komplizierten Geometrien, die mit subtraktiver Fertigung nur schwer erreichbar oder gar unmöglich wären.

Diese flexiblen Strukturen haben das gleiche Ziel wie traditionellere Mechanismen, obwohl ihnen Gelenke fehlen. Dieser Mangel an Gelenken hat mehrere Vorteile, einschließlich der Vermeidung von Reibung. Die für die Bewegung erforderliche Energie wird als elastische Energie gespeichert, die dazu verwendet werden kann, den Mechanismus in seine ursprüngliche Position zu bewegen oder in Form von Elektrizität geerntet zu werden. Darüber hinaus reduziert die damit verbundene monolithische Konstruktion die Herstellung auf den dreidimensionalen Druckprozess, wodurch eine Montage entfällt, die bei Gelenken erforderlich wäre.

Im Gegensatz zu dem gewöhnlichen Strukturentwurf sind große Deformationen, Verschiebungen und Rotationen bei nachgiebigen Mechanismen erwünscht und sind sogar die funktionale Anforderung. Daher sind gängige lineare strukturmechanische Modelle nicht mehr gültig. Neben den geometrischen Nichtlinearitäten, d.h. großen Verrückungen und großen Dehnungen, können auch nichtlineare Werkstoffeigenschaften sowie Kontakt eine Rolle spielen. Kontakt kommt vor allem in Form des Selbstkontakts sowie zwischen einer nachgiebigen Struktur und deren Werkstück oder Ladegut. Hier werden allerdings die Werkstoff- und Kontaktnichtlinearitäten bei der strukturmechanischen Analyse vernachlässigt.

Es stehen uns eine Reihe an Modellbildungen zur Verfügung und diejenigen, die hier behandelt werden, sind wie folgt:

- Starrkörpersysteme mit flexiblen Gelenken
- Balkenmodellierung mit großer Deformation

- Nichtlineare Methode der finiten Elemente

Beispiele werden jeweils für den Entwurf, die Analyse und — hier die Hauptaspekte — die Entwurfsoptimierung nachgiebiger Strukturen.

Die Optimierung ist ein mathematisches Werkzeug, das mit einem parametrisierten Modellen gekoppelt wird um nach einem optimalen Entwurf zu suchen. Die Definition des Optimalen, d.h. die Übersetzung des ingenieurtechnischen Entwurfsproblems in eine mathematische Optimierungsaufgabe, ist der erste Herausforderung. Hier wird mögliche Optimierungsformulierungen für den Entwurf nachgiebiger Strukturen eingeführt und dargestellt. Die Besonderheiten der einzelnen Modellierungsmethoden werden hinsichtlich der Systemantworten und deren Integration in die Entwurfsoptimierung erläutert.

Letztens wird die Topologieoptimierung als Methode zur Auslegung nachgiebiger Mechanismen motiviert und erklärt. Da nachgiebige Mechanismen große Verformungen aufweisen, die Notwendigkeit von der Topologieoptimierung mit der nichtlinearen Methode finiter Elemente wird gezeigt. Das hier erläuterte Topologieoptimierungswerkzeug wurde von Reinisch (2017) implementiert für geometrische Nichtlinearitäten und von Reinisch (2018) deutlich weiterentwickelt. Die Anwendung der Optimierungsmethoden werden mit Beispielen aus der Robotik dargestellt.

References

- Bruns, T. E. and D. A. Tortorelli (2003). An element removal and reintroduction strategy for the topology optimization of structures and compliant mechanisms. *International Journal for Numerical Methods in Engineering* 57, 1413–1430.
- Cao, L. (2015). *On advancing the topology optimization technique to compliant mechanisms and robots*. Ph.D. diss.
- Howell, L. L. (2001). *Compliant mechanisms*.
- Pedersen, C. B. W., T. Buhl, and O. Sigmund (2001). Topology synthesis of large-displacement compliant mechanisms. *International Journal for Numerical Methods in Engineering*, 2683–2705.
- Reinisch, J. (2017). Topologieoptimierung von Compliant Mechanisms mit nichtlinearer FEM und Spannungsrestriktionen. Semester thesis, Lehrstuhl für Leichtbau, Technische Universität München. Advisor: J. Achleitner.
- Reinisch, J. (2018). Synthesis of Compliant Mechanisms for Morphing Wings with Nonlinear Topology Optimization. Master thesis, Lehrstuhl für Luftfahrtsysteme, Technische Universität München. Advisor: J. Achleitner.
- Sigmund, O. (1997). On the design of compliant mechanisms using topology optimization. *Mechanics of Structures and Machines* (25), 493–524.
- Zentner, L. (2014). *Nachgiebige Mechanismen*. De Gruyter Oldenbourg.

Herausforderungen in Design und Kalibration für hoch-präzise Mikroroboter für Spektroskopie Messungen von Galaxien

Luzius Kronig, Dr. Philipp Hörler, Prof. Hannes Bleuler, Prof. Jean-Paul Kneib, Dr. Mohamed Bouri
(EPFL Lausanne, Laboratoire de Systèmes Robotiques)

Der Nobelpreis für Physik ging 2011 an Saul Perlmutter, Brian Schmidt und Adam Riess für ihre 1998 publizierten Ergebnisse, die zweifelsfrei eine Beschleunigung der kosmischen Expansion nachweisen. Bis heute ist die Ursache dieser Beschleunigung völlig „im Dunkeln“. Sie wird daher einer noch unbekannten „dunklen Energie“ zugeschrieben.

Ein erster Schritt, dieser dunklen Energie auf die Spur zu kommen, ist die Aufnahme einer genaueren 3-D Verteilung der fernen Galaxien. Dazu ist die Rotverschiebungs-Messung von Millionen schwach leuchtender (weil fernen) Galaxien notwendig. Ein Ausschnitt des Nachthimmels wird auf die Fokalebene eines erdbasierten Teleskops abgebildet. Um nun die Messung der Rotverschiebung durchzuführen, muss eine Glasfaser mit einer Genauigkeit von $\pm 5\mu\text{m}$ für ca. 10–15 Minuten Belichtung an die entsprechende Stelle der Fokalebene gehalten werden. Die Glasfaser leitet die eingefangenen Photonen der Galaxie weiter an einen Spektrographen, der das Lichtspektrum der Galaxie genau misst. Nach der Messung kann die Glasfaser auf eine neue Galaxie positioniert werden. Um mehrere Galaxien parallel zu vermessen wird versucht die Anzahl verwendeter Glasfasern zu maximieren.

Eine Möglichkeit, um das Platzieren von Glasfaserkabeln in der Fokalebene zu bewerkstelligen, ist der Gebrauch von SCARA-artigen Mikrorobotern. Diese Roboter haben zwei Rotationsaktuatoren für präzises Positionieren mit einem Roboterdurchmesser von nur 8-20mm je nach Teleskop. Das Positionieren der Glasfasern muss auf wenige Mikrometer genau geschehen. Normalerweise werden solche Genauigkeiten mit Hilfe eines präzisen Sensors (z.b. optische Encoder) am Wellenausgang der Aktuatoren erreicht. Für diese spezifische Anwendung mit geringen Platzverhältnissen existieren jedoch keine Sensoren mit einer genügend hohen Auflösung. Benötigt wird eine Winkelauflösung um die 0.02 grad bei einem Aktuoren-Durchmesser von nicht mehr als 4-6 mm. Die Positionsgenauigkeit wird daher mit Hilfe eines Reduktionsgetriebes realisiert. Das Reduktionsverhältnis erhöht nicht nur das Drehmoment, sondern reduziert auch Positionierfehler durch Streckung des Arbeitsbereiches zwischen Getriebeeingang und Wellenabgang. Getriebe haben jedoch verschiedene Nachteile sodass keine unendlich hohe Präzision erreicht werden kann. Die Positioniergenauigkeit ist limitiert u. a. durch ein nicht konstantes Reduktionsverhältnis, Spiel und Elastizität des Getriebes. Im Folgenden wird gezeigt wie durch Design, Regelung und Kalibrationsmethoden hohe Positioniergenauigkeiten erreicht werden können, ohne dass sperrige, hochauflösende Winkelsensoren benötigt werden.

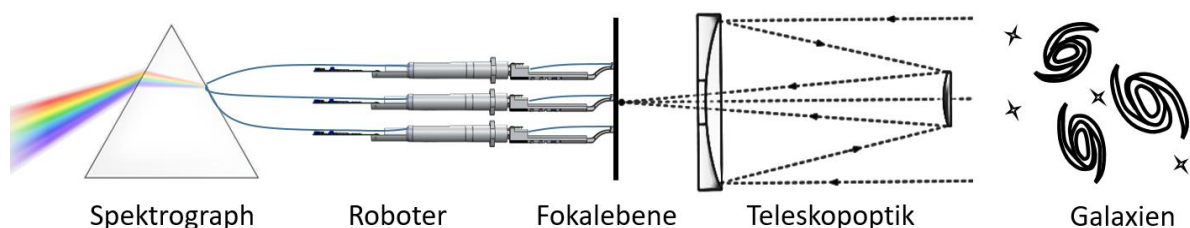


Abbildung 1: Lichtspektralmessungen von Galaxien

Design

Die Hauptanforderungen beim Design bestehen aus der absoluten Positioniergenauigkeit, der korrekten Ausrichtung der Glasfasern und der Einhaltung der maximalen Querschnittsfläche. Gute absolute Positioniergenauigkeit und Ausrichtung der Glasfasern sind unumgänglich um einen Lichtverlust zu minimisieren. Die normalerweise verwendeten Glasfasern haben einen lichtdurchlässigen Faserkern von ungefähr 100 μm wobei die zu messenden Galaxien, auf die Fokalebene projiziert, eine Grösse von ungefähr 80 bis 100 μm haben. Die x-y Positioniergenauigkeit der Fasern muss daher auf wenige Mikrometer genau sein. Ein Lichtverlust tritt auch auf durch Vignettierung. Dabei handelt es sich um den Ausrichtungsfehler der Glasfasern, welche zum gemeinsamen Fokalfunkt der Fokalebene zeigen müssen. Der Fehler der Faserausrichtung darf generell nicht grösser als 0.3 Grad betragen. Die Ausrichtung wird nicht aktiv gesteuert und muss daher durch sorgfältiges Design und Herstellung gewährleistet sein. Letztendlich ist darauf zu achten die Querschnittsfläche der Roboter klein zu halten, da die Anzahl der Fasern für eine fixe Grösse der Fokalebene gegeben ist. Die Fokalebene solcher Teleskope haben einen Durchmesser um die 1000mm mit einer Faserdichte zwischen 500 und 5000 Fasern pro Ebene. Der maximale Durchmesser p von Robotern bewegt sich daher zwischen nur 7 und 25 mm. In Zukunft wird sich die Faserdichte wahrscheinlich noch erhöhen und Roboter mit noch kleineren Querschnittsflächen werden benötigt.

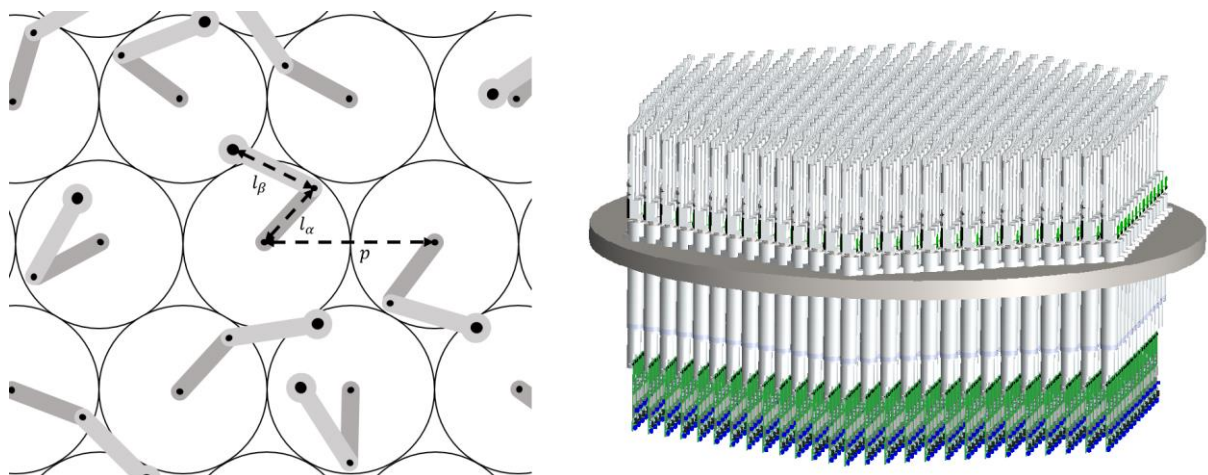


Abbildung 2 Links: Überlappung der Arbeitsbereiche, Rechts: Roboter in der Fokalebene

Abbildung 2 zeigt die Darstellung von den Robotern in der Fokalebene. Der Arbeitsraum der einzelnen Einheiten überlappt sich. Zwischen zwei Observationen werden die Glasfasern auf neue Zielgalaxien gesteuert wobei darauf geachtet werden muss, dass die Arme nicht kollidieren. Da die Roboter in Durchmesser limitiert sind, wird der zur Verfügung stehende Platz in Länge ausgenutzt. Abbildung 3 zeigt den Querschnitt eines einzelnen Roboters. Die Glasfaser ist am zweiten Arm fixiert und verläuft mittig durch die hohle Rotationsachse des ersten Arms damit eine Krümmung der Faser soweit wie möglich minimiert werden kann. Zwei Bürstenlose Elektromotoren steuern die zwei Freiheitsgrade. Beide Rotationsachsen werden ausserhalb der Elektromotoren separat gelagert um die strikten Anforderungen der präzisen Faserausrichtung zu gewährleisten. Dabei werden die Kugellager so weit voneinander versetzt damit Fertigungstoleranzen einen vernachlässigbaren Einfluss auf die richtige Achsenausrichtung haben.

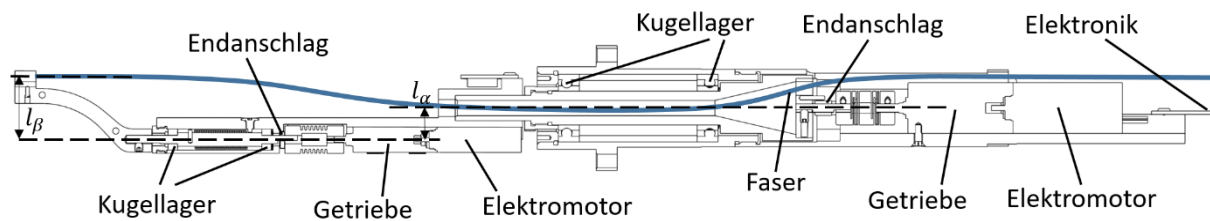


Abbildung 3 Querschnitt eines Roboters

Zusätzlich müssen die Kugellager vorgespannt werden um ein taumeln zu verhindern damit die Rotationsachsen möglichst kreisförmig drehen (siehe Abbildung 4). Während der Belichtungszeit einer Beobachtung stehen die Roboter still. Dagegen bewegt sich das Teleskop und korrigiert die Erdrotation was eine Verschiebung des Gravitationsvektors mit sich führt. Der ganze Roboter ist daher so steif wie möglich konstruiert um eine Biegung des Roboters durch Eigengewicht oder durch das Ziehen der Faser zu minimieren.

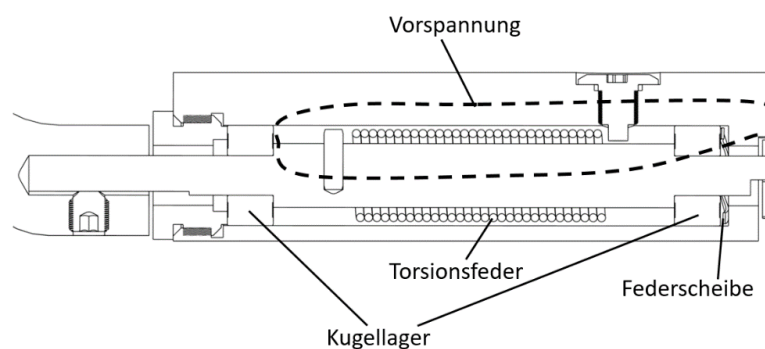


Abbildung 4: Detailansicht für die Lagerung einer Rotationsachse mit Spielreduktion

Hoch präzise Positionssteuerung wird normalerweise erreicht durch genaue Sensoren wie z.B. Interferometer. Zudem wird normalerweise versucht Coulomb Reibung zu Reduzieren um ein Stick-slip Effekt vermieden. Dies ist möglich mit flexiblen Lamellen oder mit Luft und Magnetlagern. Es ist bei dieser Anwendung jedoch wegen der geringen Platzverhältnissen weder möglich solche Aktuatoren noch Sensoren mit genügend hoher Auflösung zu verwenden. Für eine genügende Präzision werden deshalb Reduktionsgetriebe verwendet. Diese Getriebe haben jedoch gewisse Nachteile wie ein nicht konstantes Reduktionsverhältnis, Spiel und Elastizität. Diese Elemente müssen in Design und Regelung berücksichtigt werden. Das Reduktionsverhältnis des Getriebes wird so gewählt damit eine genügende Präzision erreicht wird. Zum Beispiel wird bei einer Motorencoderauflösung von 5 grad und mit einer geforderten Präzision am Wellenabgang von 0.01 grad ein Reduktionsverhältnis von mindestens 500:1 benötigt. Des Weiteren wird das Getriebe so gewählt, dass es durch die extern wirkenden Kräfte mithilfe der internen Reibung nicht rückwärtsfahrbar ist. Dies ist erforderlich da die Roboter während der Galaxienbeobachtung ausgeschaltet sind und kein Haltemoment liefern. In einem ersten Schritt wird schon während dem Design das Spiel des Getriebes soweit wie möglich reduziert. Dabei gibt es drei verschiedene Vorgehensweisen (Abbildung 5).

- Die einfachste Möglichkeit ist ein Spiel reduziertes Getriebe zu verwenden. Spiel-reduzierte Getriebe für diese Anwendungen besitzen meist zwei parallel Laufende Stirnradgetriebe die mit Hilfe eines gemeinsamen Eingangs und Abgangszahnrad gegeneinander verspannt werden. Dies sorgt für eine Torsionsspannung durch das ganze Getriebe welches das Spiel minimiert. Der Vorteil dieser Methode ist eine konstante Torsionsspannung durch das gesamte Getriebe.

- Eine zweite Möglichkeit ist die Verwendung einer Torsionsfeder. Da der Arbeitsbereich nicht grösser als eine volle Umdrehung beträgt ist es möglich am Wellenausgang eine Torsionsfeder zu verwenden. Ein Nachteil der Torsionsfeder ist neben dem limitierten Arbeitsbereich auch dass die Torsionsspannung nicht durch das ganze Getriebe aufrechterhalten werden kann. Da das Getriebe per Design nicht rückwärtsfahrbar ist, reduziert die Torsionsfeder das Spiel nicht für die allerersten Etagen des Getriebes. Eine Kombination aus Reibung und Abschwächung des Momentes durch die Untersetzung bewirkt das die Torsionsspannung nur für die letzten Zahnradetagen wirkt. Das Moment der Torsionsfeder muss grösser sein als die extern wirkenden Kräfte und Momente um sicher zu gehen, dass die Position des Getriebeabgangs nicht im Totgang zu liegen kommt.
- Die dritte Möglichkeit ist die Verwendung eines Reibungselements. Ein Element wird an den Wellenabgang gedrückt und erzeugt so Couloumb Reibung welche auch in diesem Fall höher sein muss als die extern wirkende Kräfte und Momente. Die Reibung vermindert kein Spiel an sich, sie erzeugt jedoch ein Haltemoment und verhindert so das sich die Zielposition bewegen kann während der Belichtungszeit. Der Nachteil von zusätzlichen Reibungselementen besteht in möglichem Abrieb welcher das Teleskop kontaminieren könnte.

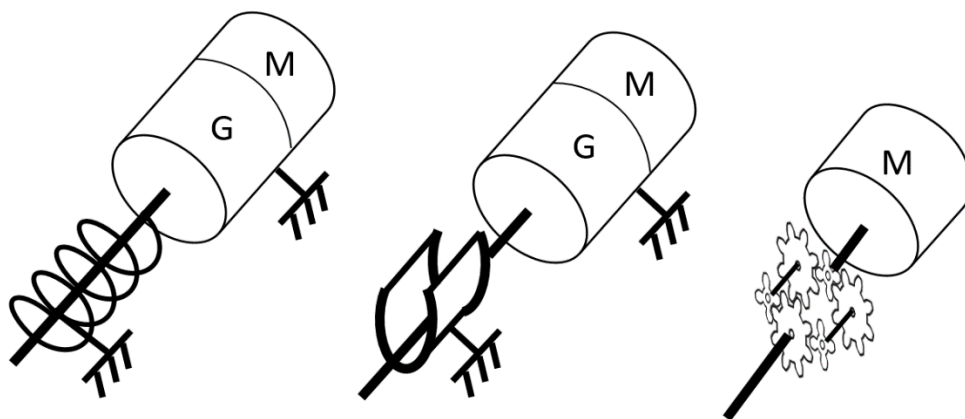


Abbildung 5 Spielreduktion Links: Torsionsfeder, Mitte: Reibungselement, Rechts: Getriebe

Encoder auf der Motorenseite werden benötigt um die exakte Position der Faser zu berechnen. Die Encoder sind jedoch nur absolut für eine Umdrehung auf Motorenseite. Um den ganzen Arbeitsbereich zu messen wird daher ein Zähler verwendet der sich erhöht oder verkleinert bei jeder Umdrehung des Motors. Als Sicherheit, falls die Anzahl Umdrehungen z.B. durch plötzlichen Stromausfall verloren geht, wird ein Initialisierungsmechanismus eingebaut. Dabei werden die Achsen des Roboters auf eine bekannte Referenzposition gefahren und der Zähler auf null gesetzt. Es gibt verschiedene Möglichkeiten für die Realisierung der Referenzposition. Die einfachste Variante ist ein mechanischer Endanschlag. Dabei wird die Rotationsachse in den Endanschlag gesteuert dabei steigt das benötigte Drehmoment des Motors abrupt an. Dieser Anstieg wird gemessen und somit ist die Roboterkonfiguration bekannt und der Zähler kann initialisiert werden. Es ist wichtig, dass das Getriebe dabei nicht beschädigt werden kann. Eine andere Variante die Referenzposition zu finden ist der Gebrauch von Optischen, Elektrischen oder Tastsensoren. Abbildung 6 zeigt zum Beispiel ein Tastschalter der ausgelöst wird sobald der Roboter auf eine Stahlkugel fährt und somit den Schalter drückt. Die benötigte Präzision für die Bestimmung der Referenzposition ist geringer

als die Verlangte Präzision am Wellenausgang, da nur die Position der Rotationsachse auf Motorenseite bestimmt werden muss. Die Repetabilität der Referenzposition muss daher kleiner sein als eine Umdrehung auf Motorenseite.

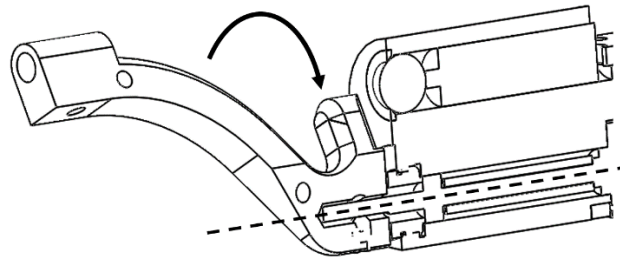


Abbildung 6 Tastschalter für die Kalibration der Referenzposition

Kalibration

Die Grundbedingung für die präzise Positionsregelung ist eine gute Positionswiederholbarkeit von nicht mehr als 0.02 Grad. Des Weiteren muss jeder einzelne Roboter kalibriert werden da z.B. die Armlängen auf wenige Mikrometer genau bekannt sein müssen. Dafür wird ein eigens dafür entwickelter Prüfstand verwendet, welcher die genaue Position und Ausrichtungswinkel der Faserspitze messen kann. Abbildung 7 zeigt die Funktionsweise des Prüfstand. Ein Roboter wird fixiert und mit einer Single-Mode-Faser ausgestattet welche mit einem Laser beleuchtet wird. Der aus der Faser tretende Lichtkegel wird mit zwei Kameras gemessen. Die erste Kamera misst die exakte x-y Position der Faserspitze und die zweite Kamera misst mit Hilfe einer Fokallinse den Austrittswinkel des Lichtkegels. Die x-y Positionsmessungen werden für die Kalibration benötigt. Mit dem Messen des Austrittswinkels wird die korrekte Faserausrichtung und somit auch die richtigen Rotationsachsenausrichtungen der Achsen überprüft.

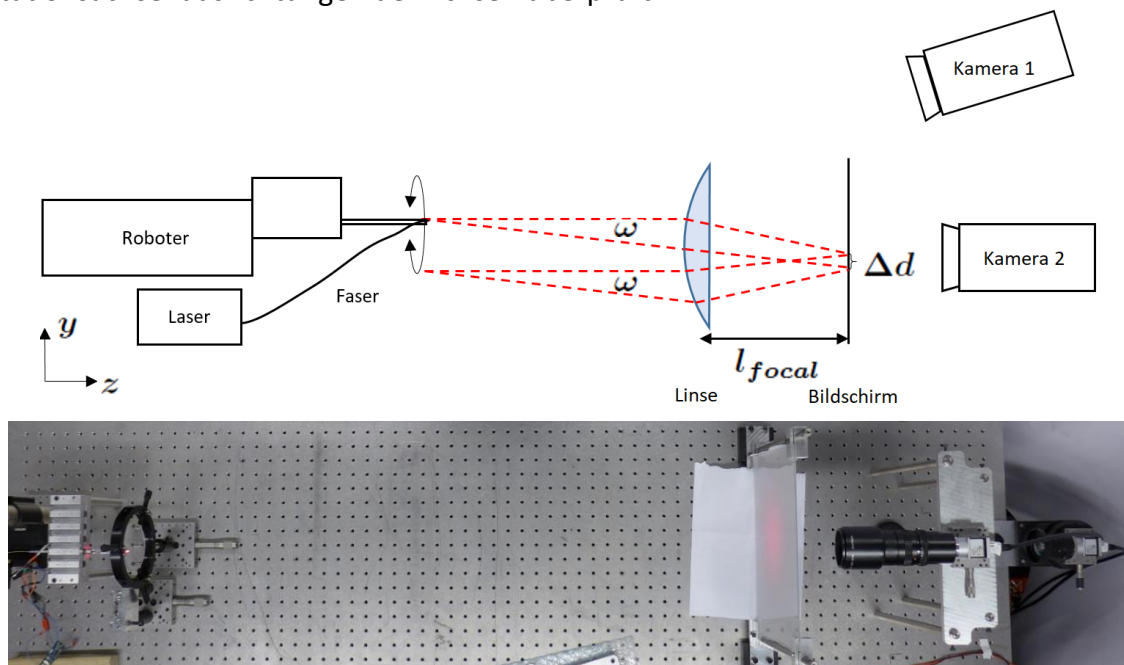


Abbildung 7 Optischer Prüfstand [6]

Für die Kalibration werden mehrere Positionsmessungen für verschiedene Motorpositionen durchgeführt. Zuerst wird die Referenzposition angefahren um die Kalibrationsmessung in der Initialposition zu starten. Dann werden die Achsen an vordefinierte Motorpositionen gesteuert wo Faserposition und Winkel gemessen wird. Die angesteuerten Motorpositionen

entsprechen eigenständigen Kreisen für beide Rotationsachsen. Entweder wird die Alpha-achse bewegt mit fixierter Beta-Achse oder Umgekehrt (Abbildung 8).

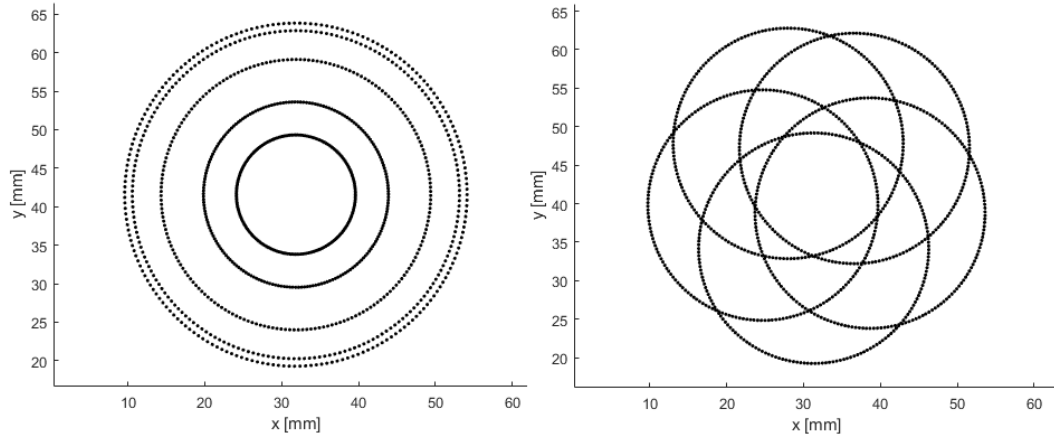


Abbildung 8 Gemessene Positionen für die Kalibration links: alpha achse dreht rechts: beta achse dreht, 200 Punkte pro Kreis

Der Roboter wird mit einem simplen Starrkörpermodell angenähert. Das Modell welches die Position der Faser r_{tot} in der x-y Ebene beschreibt wird durch die folgende Gleichung wiedergegeben und auf Bild 9 dargestellt. Wobei l_α und l_β die Armlängen sind, x_c und y_c sind die Mittelpunktsskoordinaten und die Winkel α und β sind schliesslich die Ausgangswinkel der Getriebe.

$$\begin{aligned} r_{totx} &= x_c + l_\alpha \cdot \cos(\alpha + \alpha_{off}) + l_\beta \cdot \cos(\alpha + \alpha_{off} + \beta) \\ r_{toty} &= y_c + l_\alpha \cdot \sin(\alpha + \alpha_{off}) + l_\beta \cdot \sin(\alpha + \alpha_{off} + \beta) \end{aligned}$$

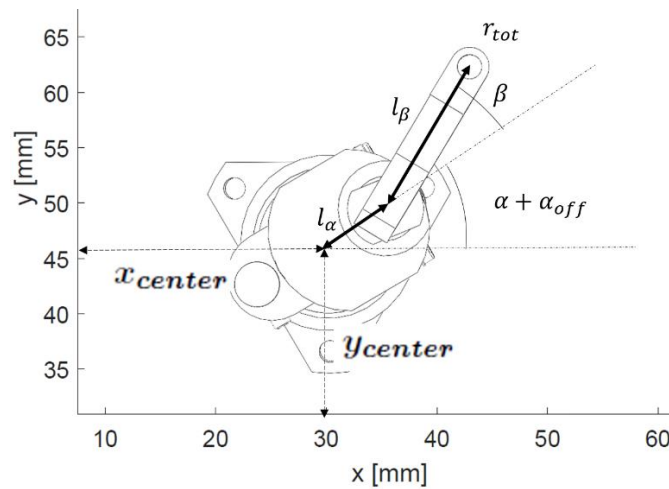


Abbildung 9 Roboter Draufsicht [6]

Die Parameter des Starrkörpermodells können mit Hilfe der gemessenen Punkte $r_{meas\ i}$ und eines numerischen Optimierungsalgorithmus gefunden werden:

$$\min_{x_c, y_c, l_\alpha, l_\beta, \alpha_{off}} \left\| \begin{matrix} r_{tot1} - r_{meas1} \\ \vdots \\ r_{toti} - r_{measi} \end{matrix} \right\|_2^2$$

Wie schon erwähnt wird die Positionsgenauigkeit mit dem Reduktionsgetriebe realisiert. Das Verhältnis zwischen Getriebeeingang α_{motor} und Ausgang α entspricht jedoch leider nicht $\alpha = k \cdot \alpha_{motor}$. Das Getriebe besitzt u.a. ein nicht konstantes Reduktionsverhältnis k , Spiel und Elastizität. Abbildung 10 zeigt ein mögliches Modell des Getriebes. Das Modell ist in

hohem Masse nicht linear und besitzt viele unbekannte Parameter welche nicht alle bestimmt und kalibriert werden können.

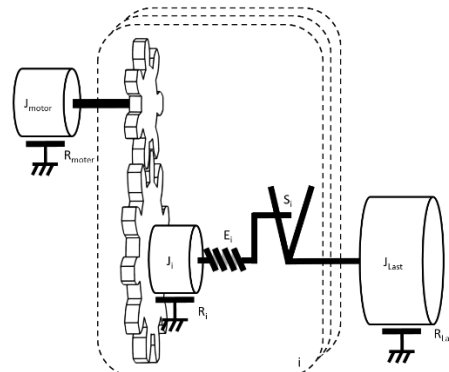


Abbildung 10 Getriebemodell

Unverzichtbar ist jedoch, dass sich ändernde Reduktionsverhältnis $k(\alpha)$ zu kalibrieren. Eine konstante Untersetzung ist gegeben, wenn die Eingriffslinie der Zahnräder eine nicht verschiebbare Gerade ist. Zwei Hauptwirkungen für ein nicht-konstantes Untersetzungsverhältnis wurden identifiziert. Zum einen besitzen die hergestellten Zahnräder nicht ein perfektes Evolventen-Zahnprofil was dazu führt, dass die Eingriffslinie von einer Geraden abweicht. Ausserdem kann jede Zahnpaarkombination ihre eigene verschiedene Eingriffslinie besitzen. Die zweite Ursache für eine sich ändernde Untersetzung ist das der Grundkreisradius eines Zahnrades nicht konstant bleibt über eine volle Umdrehung. Ursachen sind Fertigungsfehler welche sich in einer wobbelnde (eiernde) Rotationsachse des Zahnrades oder nicht kreisrunden Zahnräder manifestieren. Die Folge ist, dass sich die Eingriffslinie während einer Umdrehung verschiebt.

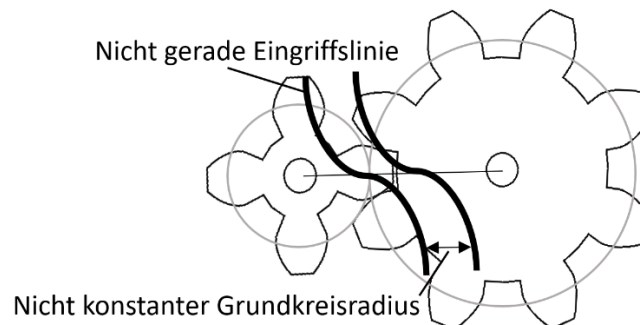


Abbildung 11 Effekte für ein nicht konstantes Reduktionsverhältnis

Abbildung 11 zeigt eine Skizze der beiden Effekte. Das Untersetzungsverhältnis k hängt somit von der Motorenposition α und besteht aus der Summe der zwei Effekte für alle Zahnräder des Getriebes. Die nicht konstante Untersetzung wird als die Abweichung von der Durchschnittsuntersetzung kalibriert und wird im Folgenden als Untersetzungskorrektur benannt:

$$\alpha = k \cdot \alpha_{motor} + \delta(\alpha)$$

$$\beta = k \cdot \beta_{motor} + \delta(\beta)$$

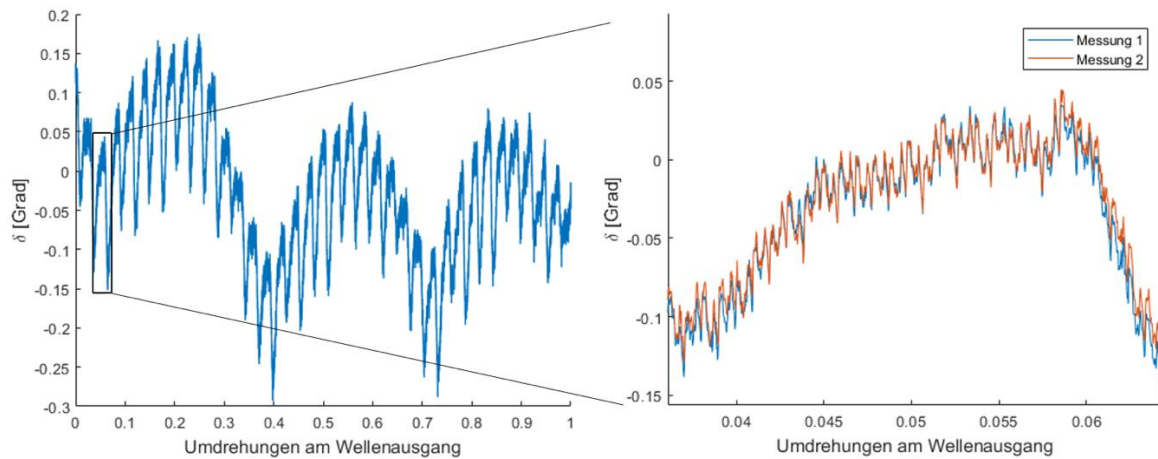


Abbildung 12 Untersetzungskorrektur eines Planetengetriebes für eine Umdrehung am Wellenausgang

Für eine ganze Umdrehung des Getriebeabgangs dreht sich jedes Zahnrad n mal und somit wiederholt sich der Effekt eines nicht runden Grundkreisradius auch n mal. Unter Annahme, dass sich die Eingriffslinien eines Zahnpaars ähneln wiederholt sich der Effekt des Zahnprofils $n \cdot z$ mal für eine Umdrehung am Getriebeabgang wobei z die Anzahl Zähne pro Zahnrad ist. Die Amplitude der Untersetzungskorrektur für jedes Zahnrad wird kleiner je weiter es vom Getriebeabgang ist, da die Amplitude durch das Untersetzungsverhältnis der nachfolgenden Zahnräder verkleinert wird. Bei konstanten Fertigungstoleranzen ist daher der Einfluss des letzten Zahnrades am grössten. Die Untersetzungskorrektur wird für jede Achse einzeln bestimmt, da für die Kalibration eigenständige Kreise abgefahren werden. Abbildung 12 zeigt als Beispiel die gemessene Untersetzungskorrektur von einem 6mm Planetengetriebe. Der Einfluss der 36 Zähne des letzten Zahnrades sind leicht zu erkennen. Es ist zu beachten, dass die Untersetzungskorrektur bis zu 0.3 Grad betragen kann und daher unbedingt kalibriert werden muss um die gewünschte Genauigkeit von 0.05 Grad zu erreichen. Mithilfe einer Frequenzanalyse lässt sich eruieren welches Zahnrad wie viel zur Untersetzungskorrektur beiträgt (Abbildung 13). Auf der Abbildung ist zu sehen wie die Amplitudenspitzen den jeweiligen Zahnrädern zugewiesen werden können.

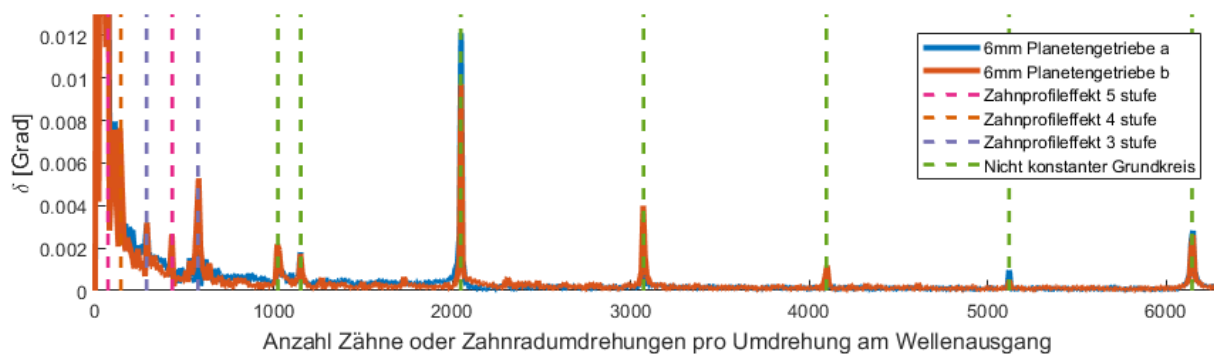


Abbildung 13 Frequenzanalyse der Untersetzungskorrektur

Wenn man den Elektromotor an eine gewisse Encoder-Position steuert und dann die Position des Wellenabgangs misst, so stellt man fest, dass die Position nicht nur von der aktuellen Encoder-Position abhängt, sondern auch von dessen Anfahrtsverlauf. Dies kann man sich einfach durch das Spiel im Getriebe erklären. Das Spiel besteht auch dann, wenn das Haltemoment durch Reibung oder Torsionsfeder die Ausgangsposition konstant hält, da ein Restspiel in den ersten Untersetzungsstufen des Getriebes verbleibt. Hier wird der Begriff Hysteresis verwendet für den Winkelunterschied am Ausgang für eine Motorenposition mit entgegengesetzten Anfahrtswegen. Um die gewünschte Wiederholgenauigkeit zu erreichen

muss daher darauf geachtet werden, dass die Anfahrwinkel der Motoren eine monoton steigende (oder fallende) Funktion besitzen und der Anfahrswinkel grösser als die maximale Hysteresis ist. Hysteresis besteht jedoch auch dann noch wenn ein Getriebe mit kompletter Spielreduktion benutzt wird (Abbildung 5 rechts). Der Grund hierfür liegt an einem Zusammenspiel von Reibung und Elastizität wie auf Abbildung 14 ersichtlich ist.

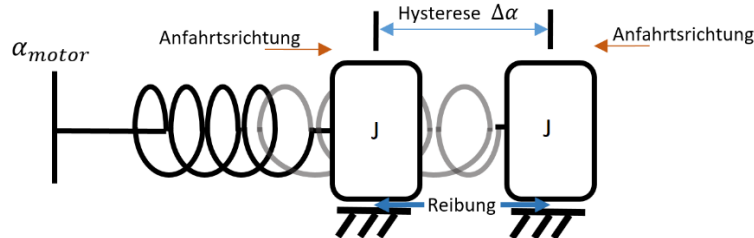


Abbildung 14 Hysteresis durch Elastizität und Reibung

Wie beim Untersetzungsverhältnis hängt die Hysteresis von der Position der Rotationsachse ab. Die Elastizität zwischen Getriebeeingang und Ausgang hängt zum Beispiel davon ab ob wie die Zahnräder ineinandergreifen. Abbildung 15 zeigt die gemessene Hysteresis eines Spielreduzierten 6mm Getriebes. Die Elastizität sowie die Untersetzungskorrektur sind wiederholbar. Die Reibung ist jedoch abhängig von verschiedenen Faktoren wie Drehgeschwindigkeit, Schmierung, Temperatur usw. Es ist daher essentiell die Torsionssteifigkeit des Getriebe so hoch wie möglich zu halten um den Einfluss der Reibung und externen Drehmomente auf die Hysteresis so klein wie möglich zu machen. Höhere Steifigkeit kann z.B. durch eine adäquate Materialauswahl oder Härtebehandlung der Zahnräder erreicht werden.

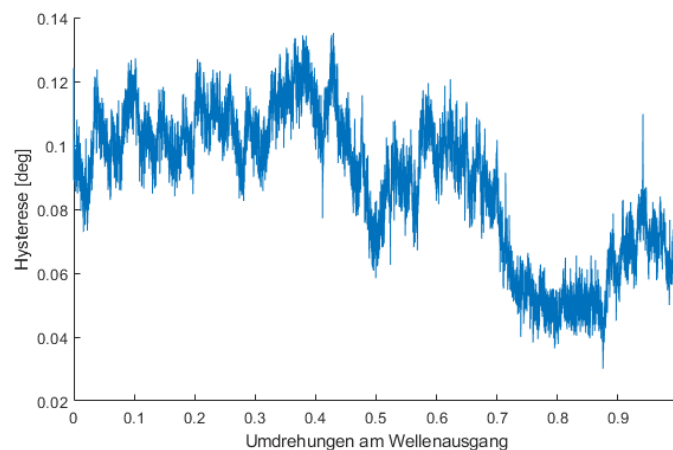


Abbildung 15 Hysteresis für ein 6mm Planetengetriebe mit Torsionsfeder

Regelung

Jede Rotationsachse wird mit einem 3 Phasen Bürstenlosen Gleichstrommotor betrieben. Jede der drei Phasen wird mit einer Pulsdauermodulation (PWM) so angesteuert, dass ein rotierendes elektrisches Feld entsteht. Der Permanentmagnet des Rotors richtet sich nach dem Magnetfeld aus. Die Einschaltdauer eines Zyklus wird wie folgt berechnet:

$$\begin{aligned} PH_A &= |v_{field}| \cdot 0.5 \cdot (1 + \sin(\alpha_{field})) \\ PH_B &= |v_{field}| \cdot 0.5 \cdot (1 + \sin(\alpha_{field} + 120^\circ)) \\ PH_C &= |v_{field}| \cdot 0.5 \cdot (1 + \sin(\alpha_{field} + 240^\circ)) \end{aligned}$$

Wobei $|v_{field}|$ die Stärke des Magnetfeldes ist und α_{field} seine Ausrichtung (siehe Abbildung 16 links). Das PWM Signal wird somit in t_{feld} und t_{halt} eingeteilt (Abbildung 16 rechts). Die Länge der Wartezeit t_{halt} regelt die Stärke des Magnetfeldvektors v_{field} .

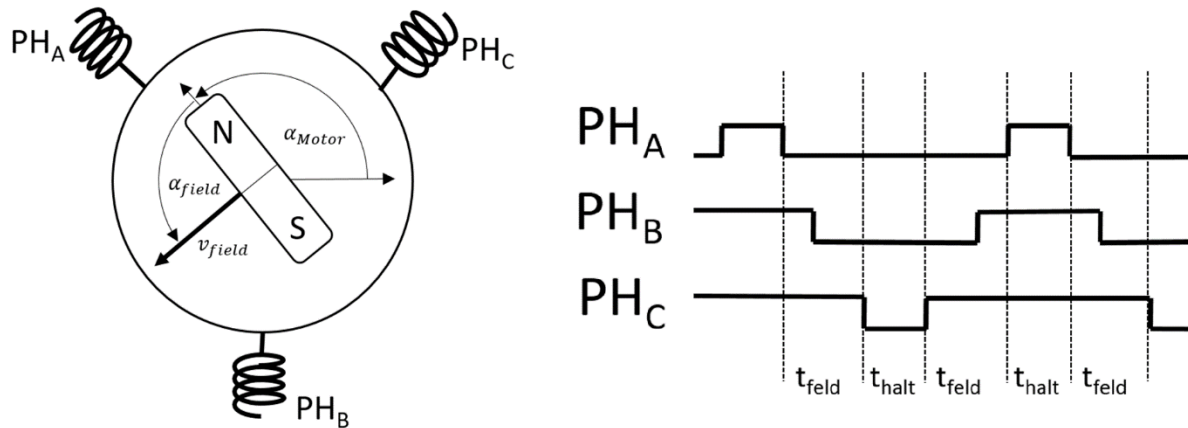


Abbildung 16 Vektorsteuerung von Bürstenlosen Gleichstrommotoren

Die Motoren können nun entweder in «open» oder «closed loop» angesteuert werden. Mit «open loop» Steuerung richtet sich der Magnet des Motors stets dem Magnetfeldvektor aus und es wird angenommen das $\alpha_{field} \approx \alpha_{motor}$. Dies funktioniert gut da die Massenträgheit auf der Motorenseite sehr klein ist und da das Drehmoment M welches auf den Motor wirkt vom Winkel zwischen generierten Feld und Magnet abhängt $M = v_{field} \times v_{motor}$. Ein Nachteil der «open loop» Steuerung ist, dass sehr viel Energie in Wärme verloren geht. Für eine ideale Steuerung beträgt die Winkeldifferenz zwischen Magnet und generiertem Feld 90 Grad. Bei der «closed loop» Steuerung wird diese optimale Winkeldifferenz mit Hilfe eines Encoders eingehalten. Dazu wird ein simpler PID Regler verwendet welcher die Differenz zwischen Regelungs-Sollwert α_{motor} und gemessene Ausrichtung $\tilde{\alpha}_{motor}$ minimiert:

$$M = f_{PID}(\alpha_{motor} - \tilde{\alpha}_{motor})$$

$$|v_{field}| = |M|$$

$$\alpha_{field} = \alpha_{motor} + \text{sign}(M) \cdot 90^\circ$$

Jeder Roboter erhält eine vorausberechnete Bewegungsbahn $r_{tot}(t)$ welche dann mit der inversen Kinematik in Rotationssteuerbefehle ($\alpha(t)$, $\beta(t)$) umgerechnet wird. Bevor diese mit dem PID Regler abgefahren werden kann, muss sie noch in Koordinaten auf der Motorenseite umgewandelt werden mit Berücksichtigung der Untersetzungs-korrektur:

$$\alpha_{motor} = 1/k (\alpha - \delta(\alpha))$$

$$\beta_{motor} = 1/k (\beta - \delta(\beta))$$

Bei der Erstellung der Bewegungsbahn ist darauf zu achten, dass es zu keinen Kollisionen zwischen den einzelnen Robotern kommt. Ausserdem muss die Bahn so berechnet werden, dass die letzte Annäherungsstrecke bevor Stillstand die Anfahrtsrichtung nicht ändert um nicht im Spiel der Hysterese zu stoppen. Dabei wird eine monoton steigende (oder fallende) Anfahrtsstrecke gewählt. Für die «closed loop» Steuerung werden die PID Regelparameter so berechnet, dass das resultierende System überdämpft ist und es somit zu keiner Überschwingung kommen kann.

Zusammenfassung

Die Erforschung der dunklen Energie führt zu interessanten Problemstellungen in der Mechatronik, um präzise spektroskopische Messungen durchzuführen. Kleine SCARA Roboter positionieren Glasfasern mit Mikrometer-Genauigkeit in der Fokalebene eines Teleskopes. Design, Kalibration und Regelung müssen sorgfältig durchgeführt werden um die Anforderungen zu erfüllen. Dabei müssen die Auswirkungen von Fertigungstoleranzen, Elastizität, Hysterese und Spiel beachtet und minimiert werden.

Literatur

- [1] Perlmutter, S. et al: *Measurements of Ω and Λ from 42 High-Redshift Supernovae*. The Astrophysical Journal, Volume 517, Number 2. (1999), S. 565– 586.
- [2] Fahim N.: *An 8-mm diameter Fiber Robot Positioner for Massive Spectroscopy Surveys*, Monthly Notices of the Royal Astronomical Society, vol. 450, no. 1, pp. 794–806, Apr. 2015.
- [3] Schubnell M et al: *The DESI fiber positioner system*, Proc.SPIE (2016)
- [4] Hörler P et al: *A 24mm diameter fibre positioner for spectroscopic surveys*, Proc.SPIE (2016)
- [5] Hörler P et al: *High density fiber positioner system for massive spectroscopic surveys*, Monthly Notices of the Royal Astronomical Society 2018
- [6] Kronig L. et al: *Design and performances of an optical metrology system to test position and tilt accuracy of fiber positioners*, Proc.SPIE (2018)
- [7] Jenni L. et al: *Developing micro DC-brushless motor driver and position control for fiber positioners*, Proc. SPIE 9147 (2014), S. 914744–914744–7.
- [8] Makarem L., et al: *Collision-free motion planning for fiber positioner robots: discretization of velocity profiles*. Proc. SPIE 9152 (2014), 91520Q–91520Q– 10.
- [9] Makarem L., et al: *Collision avoidance in nextgeneration fiber positioner robotic systems for large survey spectrographs*, A&A, Volume 566, no. A84, June 2014
- [10] Cirasuolo M., et al: *MOONS: the Multi-Object Optical and Near-infrared Spectrograph for the VLT*, 2014, vol. 9147, p. 91470N–91470N–13.
- [11] Fisher C., et al: *Developing engineering model Cobra fiber positioners for the Subaru Telescope's prime focus spectrometer*, 2014, vol. 9151, p. 91511Y–91511Y–13.
- [12] McConnachie A., et al: *The Maunakea Spectroscopic Explorer: the science driven design rationale* SPIE Astronomical Telescopes + Instrumentation. International Society for Optics and Photonics, 2014.

Entwicklung einer Wechselschnittstelle für Tools in Nanofabrikationsmaschinen mit hochreproduzierbarer Lagedefinition

Development of a changing interface for tools in nanofabrication machines with a highly reproducible position definition

Florian Weigert, Technische Universität Ilmenau, Institut für Maschinen und Gerätekonstruktion, 98693 Ilmenau, Deutschland, florian.weigert@tu-ilmenau.de

René Theska, Technische Universität Ilmenau, Institut für Maschinen und Gerätekonstruktion, 98693 Ilmenau, Deutschland, rene.theska@tu-ilmenau.de

Kurzfassung

Im Stand der Forschung sind Wechselschnittstellen für Sensoren und Werkzeuge bereits in unterschiedlichen Maschinenklassen zu finden. Diese erfüllen jedoch nicht die geforderten höchsten Ansprüche an die Reproduzierbarkeit der Position für den Einsatz in Nanofabrikationsmaschinen. Kinematische Kopplungen zeigen aufgrund ihrer eindeutigen, kinematisch bestimmten Lagedefinition und einer hohen Reproduzierbarkeit eine gute Eignung für den Einsatz als Wechselschnittstelle in Nanofabrikationsmaschinen. Anhand von analytischen Berechnungen werden die wirkenden Kräfte und Verformungen der Koppelpunkte ermittelt. Für die Integration einer Justierung werden neue Anordnungen einer solchen kinematischen Kopplung untersucht. Da analytische und numerische Modelle die Reproduzierbarkeit nicht ausreichend genau abbilden können, wird ein geeigneter Messaufbau entwickelt. Dessen Eignung wird anhand einer dreidimensionalen, vektorbasierten Unsicherheitsanalyse nachgewiesen. Anschließend wird mittels der Ergebnisse der untersuchten Prototypen ein Modell zur Beschreibung des Zusammenhangs zwischen Konstruktionsmerkmalen und Reproduzierbarkeit entwickelt.

Abstract

In the state of research, changing interfaces for sensors and tools can already be found in different machine classes. However, these do not meet the highest requirements regarding the reproducibility of the position for use in nanofabrication machines. Kinematic couplings, due to their clear, kinematically determined position definition and high reproducibility, are well suited for use as changing interfaces in nanofabrication machines. On the basis of analytical calculations, the effective forces and deformations at the coupling points are determined. For the integration of an adjustment, new arrangements of such a kinematic coupling are investigated. Since analytical and numerical models cannot describe the reproducibility with sufficient accuracy, a suitable measurement setup is developed. Its suitability is proven by a three-dimensional, vector-based uncertainty analysis. Subsequently, the results of the examined prototypes will be used to develop a model for the description of the correlation between design features and reproducibility.

1 Einleitung

Dieser Beitrag befasst sich mit der Entwicklung von konstruktionswissenschaftlich fundierten Grundlagen für Wechselschnittstellen von Sensoren und Werkzeugen in Nanofabrikationsmaschinen (nachfolgend als Tools bezeichnet). Dabei ist die grundlegende Forderung eine kinematisch eindeutige Lagedefinition zwischen Tool und Maschine. Die in Ilmenau entwickelten Nanopositionier- und Nanomessmaschinen erfüllen bereits Anforderungen an eine subnanometergenaue Messung und Positionierung (NMM-1 und NPMM-200). In aktuellen Forschungsarbeiten werden diese Maschinen für die Nanofabrikation erweitert. Dadurch entsteht vermehrt die Forderung nach einer hochreproduzierbaren Toolwechselschnittstelle. Der in diesem Beitrag vorgestellte Ansatz, basierend auf einer kinematischen Kopplung, verspricht eine erzielbare Reproduzierbarkeit der Position von weniger als 100 nm.

2 Stand der Forschung

Im Stand der Forschung sind Wechselschnittstellen in unterschiedlichen Maschinenklassen zu finden. Die in Werkzeugmaschinen vertretenen Wechselschnittstellen werden in diesem Beitrag nicht näher betrachtet, da diese überwiegend für rotierende oder rotationssymmetrische Tools zum Einsatz kommen und dort keine Definition der Drehlage um die Toolachse erforderlich ist. In Koordinatenmessmaschinen kommen häufig Wechselschnittstellen zum Einsatz, die als kinematische Kopplung ausgeführt sind. Diese Wechselschnittstelle dient dazu, die Tasteranordnung über die Tasteraufnahme bezüglich Geometrie und räumlicher Ausrichtung messaufgabenspezifisch einzuwechseln. In einigen Fällen ist auch eine Informationsübertragung über elektrische Kontakte vorgesehen, wobei die Wirkung der auftretenden parasitären Kräfte als vernachlässigbar angenommen wird.

Für die NMM-1 existiert aus vorangegangenen For-

schungsprojekten bereits ein erster Ansatz eines Toolwechselsystems. Bei diesem Prinzip wird ein konventioneller Objektrevolver aus einem Mikroskop verwendet, mit dem jedoch keine ausreichend hohe Reproduzierbarkeit erzielt werden konnte [4]. Kompensationsmöglichkeiten, die auf der Messung von Marken auf dem Messtisch beruhen sind für die Nanofabrikation ungeeignet, da die Fabrikationstools nicht immer ein eigenes Messsystem besitzen und die Anbringung solcher Marken nicht generell möglich ist. Kinematische Kopplungen zeichnen sich theoretisch durch eine Lagedefinition frei von Überbestimmtheiten aus. Die Umsetzung der kinematischen Kopplung geht in der Realität mit unvermeidbaren Gestaltabweichungen einher, die sich in Form von Abweichungen in der Reproduzierbarkeit der Position zeigen. Ein weiterer Einfluss ist durch die Kontaktreibung gegeben. Die bekanntesten Ausführungsformen kinematischer Kopplungen werden als Kelvin-Anordnung und Maxwell-Anordnung bezeichnet. Die Kelvin-Anordnung besteht aus drei Kugeln die jeweils mit einem Tetraeder, einer V-Nut und einer Ebene gekoppelt sind. Bei der Maxwell-Anordnung sind die drei Kugeln hingegen mit drei V-Nuten im Winkel von 120° gekoppelt (vgl. Abbildung 1).

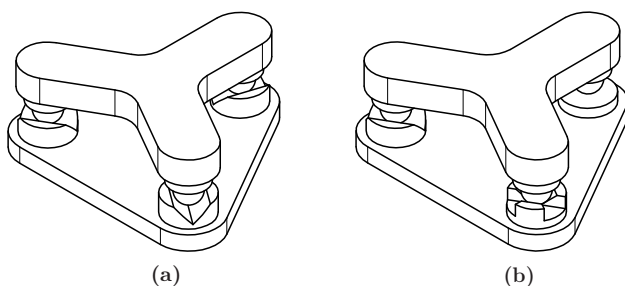


Bild 1 Kinematische Kopplungen: (a) Maxwell-Anordnung, (b) Kelvin-Anordnung

Ende der 90er Jahre wurden durch die Forschergruppe um Prof. Slocum am Massachusetts Institute of Technology Grundlagenuntersuchungen zu kinematischen Kopplungen nach der Maxwell-Anordnung durchgeführt [5, 6]. Mit Kugeln aus Siliziumnitrid und V-Nuten aus gehärtetem Stahl konnte nach einer Einlaufphase von 50 Zyklen unter Verwendung eines Schmierstoffs eine Reproduzierbarkeit von $0.33 \mu\text{m}$ in radialer und $0.30 \mu\text{m}$ in axialer Richtung erzielt werden. Als Verkippungen ergaben sich $1.6 \mu\text{rad}$ um die x-Achse und $2.8 \mu\text{rad}$ um die y-Achse.

In Erweiterung klassischer Wechselschnittstellen in Werkzeugmaschinen wurde in [3] eine hochreproduzierbare Wechselschnittstelle inklusive Energie- und Informationsübertragung für kleine Werkzeugmaschinen entwickelt. Die entwickelte Kopplung besitzt Abmessungen von $45 \text{ mm} \times 45 \text{ mm} \times 30 \text{ mm}$ und basiert ebenfalls auf einer kinematischen Kopplung mit der Maxwell-Anordnung. Die damit erreichte Reproduzierbarkeit liegt im Bereich von $0.18 \mu\text{m}$ in Translation und $9.50 \mu\text{rad}$ in Rotation. Der Fokus der zitierten Arbeit liegt auf der Entwicklung eines schaltbaren Magnetsystems zur Realisierung einer einstellbaren Vorspannkraft und einer Energie- und Informationsübertragung inklusive Messdatenverarbeitung. Eine aus-

föhrliche Betrachtung der konstruktiven Gestaltung, der Langzeitstabilität und weiterer Einflussfaktoren fehlt.

Anhand der Analyse des Forschungsstandes wird deutlich, dass besonders im Bereich der hochreproduzierbaren Wechselsysteme noch erheblicher Forschungsbedarf besteht. Die bisher erreichte Reproduzierbarkeit ist nicht ausreichend für die Nanofabrikation und in keiner der vorausgegangenen Arbeiten wurden Langzeitstabilität und Kriecheffekte untersucht. Eine ausreichende Langzeitstabilität ist schon für den Einsatz in der Nanopositionier- und Nanomessmaschine entscheidend und wird durch deren Einsatz für die Nanofabrikation vermehrt gefordert.

3 Gestaltung und Aufbau der Wechselschnittstelle

Analog zum Stand der Technik orientiert sich die Gestaltung der Wechselschnittstelle an kinematischen Kopplungen. In Ergänzung der bereits gut beschriebenen Maxwell-Anordnung wird die Kelvin-Anordnung einer genaueren Betrachtung unterzogen. Sie besitzt die vorteilhafte Eigenschaft eines definierten Drehpunktes in Form der durch einen Tetraeder und eine Kugel gebildete Koppelstelle (vgl. Abbildung 1 (b)). Nachteilig hingegen ist ein unsymmetrisches Verhalten der Wechselschnittstelle bei thermischen Einflüssen. Bei der Maxwell-Anordnung kompensieren sich globale thermische Einflüsse aufgrund des symmetrischen Aufbaus.

Für die Implementierung der Wechselschnittstelle in die Nanofabrikationsmaschine muss der Arbeitspunkt des verwendeten Tools im Abbe-Punkt der Nanofabrikationsmaschine liegen. Die Längenmessung in der Maschine erfolgt unter Einhaltung des Abbeschen Komparatorprinzips in allen Raumachsen, um Fehlereinflüsse erster Ordnung zu vermeiden. Dabei treffen sich die Längenmessachsen virtuell in Toolarbeitspunkt. Aufgrund von Fertigungs- und Montageabweichungen sowie Verschleiß kann der zulässige Bereich der Lageabweichung des Toolarbeitspunkts überschritten werden. Dieser ist für die NPM-200 mit $\pm 0.05 \text{ mm}$ angegeben [1]. Dadurch kann sich die Notwendigkeit einer Justierung ergeben. Durch eine geschickte Wahl der Anordnung der Koppelpunkte wird eine unabhän-

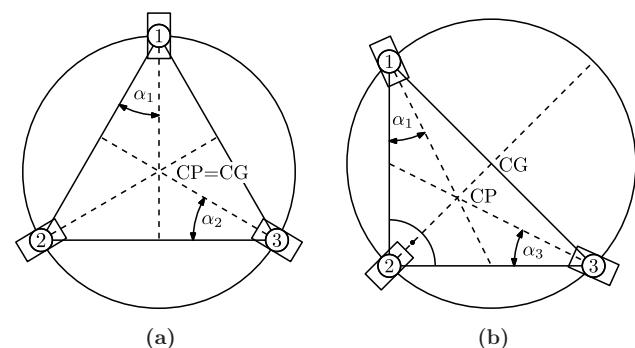


Bild 2 Unterschiedliche Anordnungsvarianten der Koppelpunkte von allgemeinen kinematischen Kopplungen mit Koppelmittelpunkt (CP), Schwerpunkt (CG) und Winkel bei Verwendung von V-Nuten (α)

gige Justierung einzelner Freiheiten und damit eine systematische Justierung möglich. Mit der in Abbildung 2 (b) dargestellten Anordnungen kann die kinematische Koppelung um eine unabhängige Justierung in drei Freiheiten erweitert werden. Anhand der Ausrichtung der richtungsabhängigen V-Nuten lässt sich weiterhin der Schwerpunkt beeinflussen, wie durch die Winkel α in Abbildung 2 dargestellt.

4 Analytische Betrachtungen der Wechselschnittstelle

In einem ersten Schritt werden analytische Berechnungen der Wechselschnittstelle durchgeführt. Dazu wurde ein Berechnungsprogramm zur Kräfteanalyse entwickelt. Eingangsgrößen sind die Größe und Richtung der eingeleiteten Kräfte, die Anordnung der Koppelpunkte und die Gestalt der Koppelstellen. Daraus werden die wirkenden Kräfte auf die Koppelpunkte und die dazugehörigen Kraftrichtungen berechnet. Anhand dieses Programms kann auch eine Bestimmung des optimalen Punkts der Krafteinleitung durchgeführt werden, um möglichst identische Kraftverhältnisse an allen sechs Koppelpunkten zu schaffen.

Anhand der ermittelten Kräfte können mittels verschiedenen Verfahren die elastischen oder möglicherweise plastischen Verformungen an den Koppelpunkten berechnet werden. Dafür eignet sich besonders die Methode der Dimensionsreduktion nach Valentin Popov. Bei diesem Verfahren wird der Kontaktbereich durch eine eindimensionale elastische Bettung bestehend aus linear elastischen Federelementen mit sehr kleinem Abstand ersetzt. Dadurch können auftretende Verformungen exakt berechnet werden. Bei dem Einsatz in der Nanofabrikationsmaschine sind Bearbeitungskräfte vernachlässigbar. Die wirkenden Kräfte auf die Koppelpunkte setzen sich aus Vorspannkraft und Massenträgheitskräften zusammen und liegen in Summe voraussichtlich unter 50 N. Die Vorspannkraft ist notwendig um mögliche Oberflächeneffekte wie Schmier- oder Flüssigkeitsfilme zu überwinden. Eine daraus resultierende Verformung am Kontaktpunkt liegt im Sub-Mikrometerbereich und ist somit vernachlässigbar. Auftretende Spannungen liegen weit unter der Fließgrenze der potenziell geeigneten Materialien. Ein direkter Zusammenhang zwischen elastischer Verformung und Reproduzierbarkeit der Position der kinematischen Kopplung ist nicht bekannt, weshalb die Größe der Verformung im weiteren Verlauf der Entwicklung keine Relevanz besitzt.

5 Entwicklung eines Messaufbaus

Die maßgeblichen Einflussgrößen auf die Reproduzierbarkeit der Position der Wechselschnittstelle wie beispielsweise Reibung lassen sich durch analytische und numerische Modelle nicht ausreichend genau abbilden. Aus diesem Grund ist hierfür ein geeigneter Aufbau zur Messung der Reproduzierbarkeit der Wechselschnittstelle in fünf Freiheiten notwendig. Für die in der NPM-200 verwendeten messenden Tools und die sich in Entwicklung befindenden Fabrikationstools besitzen Rotationen um die z-Achse

einen vernachlässigbaren Einfluss (vgl. Abbildung 3).

Die im Stand der Technik durchgeführten Untersuchungen von Wechselschnittstellen wurden mittels linearvariabler Differentialtransformatoren (LVDT) [6] und Halbleiterinterferometern [3] durchgeführt. Beide Verfahren sind für die angestrebten Messunsicherheiten im einstelligen Nanometerbereich nicht geeignet. LVDTs als Messmittel eignen sich nicht wegen ihrer zu geringen Auflösung und einer nicht kontaktfreien Messung. Halbleiter-Interferometer wiederum erfüllen nicht die Anforderungen, um auch die Langzeitstabilität der Koppelstellen mit ausreichender Genauigkeit zu untersuchen.

Die Entwicklung des Messaufbaus basiert auf dem Entwicklungsprozess nach Ilmenauer Schule. Zunächst wurden insgesamt 25 potenziell geeigneten Längen- und Winkel-Messmittel zusammengetragen. Anhand von Eigenschaften wie unter anderem der erreichbaren Auflösung, dem maximalen Messbereich, Wiederholgenauigkeit, Kontaktfreiheit und nach absolutem oder relativem Messprinzip erfolgte ein Vergleich und eine Vorauswahl. Für die Messung der geforderten fünf Freiheiten wurden die potenziell geeigneten Messmittel kombiniert und bewertet. Dabei kann die Messung der rotatorischen Reproduzierbarkeit sowohl direkt durch Winkelmessung als auch indirekt über Triangulation bestimmt werden.

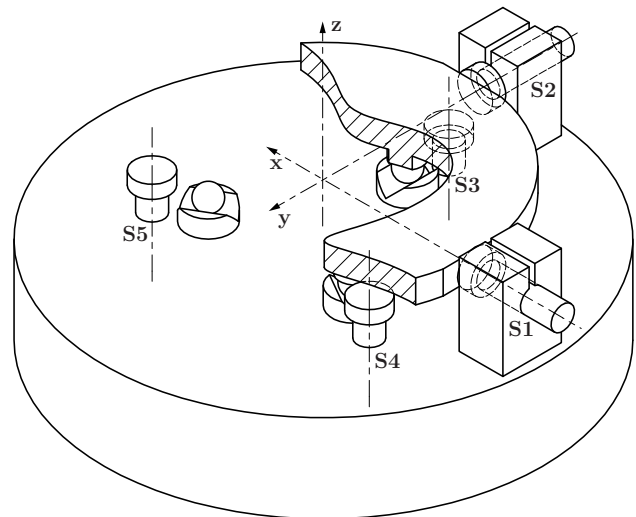


Bild 3 Technisches Prinzip des Messaufbaus mit kapazitiven Sensoren (S1-S5), Sensorbefestigungen, Oberseite und Unterseite der kinematischen Kopplung

Als Ergebnis zeigte die Kombination aus fünf kapazitiven Sensoren die beste Eignung. Diese zeichnen sich durch ein absolutes und kontaktfreies Messprinzip, Auflösungen im Subnanometerbereich und eine hochgradig lineare Kennlinie aus. Das technische Prinzip des Messaufbaus ist in Abbildung 3 dargestellt. Für eine geringe Messunsicherheit sollte der metrologische Kreis möglichst kurz geschlossen werden. Aus diesem Grund wurden die Sensoren nahe bei den Koppelpunkten angeordnet. Gleichzeitig sollte der Abstand der Sensoren S3, S4 und S5 für eine hohe Auflösung der Rotation möglichst groß sein. Um weitere störende Einflüsse auszuschließen wird der Messaufbau in einer Vakuumklimakammer betrieben. Darin kann die Temperatur

mittels eines Käte-Umwälzthermostats auf $\pm 0,01\text{K}$ genau reguliert werden. Das Vakuum wird analog zur NPMM-200 auf 1 mbar eingestellt [1]. Zusätzlich ist die Kammer pneumatisch aktiv schwingungsgedämpft.

5.1 Unsicherheitsbetrachtung

Zum Nachweis der Eignung des in Abbildung 3 dargestellten Messaufbaus wurde eine Unsicherheitsbetrachtung durchgeführt. Diese basiert auf dem dreidimensionalen Vektor-Ansatz, mit dem auch die NPMM-200 beschrieben wurde. Grundlage dieses Ansatzes ist die Modellbildung mittels einfacher, erweiterbarer Untermodelle die als Vektor einzelne Einflüsse beschreiben. Diese werden am Ende zu einem geschlossenen Vektorzug addiert. Bei der Modellbildung wird die Analogie einer kinematischen Koppelung und eines Positioniersystems mit sechs Freiheiten verwendet. Zu zwei unterschiedlichen Zeitpunkten wird ein geschlossener Vektorzug gebildet. Die Differenz der Bezugskoordinatensysteme der beiden Vektorzüge stellt die Messunsicherheit dar.

Die Ergebnisse der Unsicherheitsbetrachtung zeigen, dass vor allem thermische Einflüsse kritisch sind. Bei der Verwendung von konventionellen Konstruktionswerkstoffen wie Aluminium oder Stahl und einer Temperaturstabilität von $\pm 0,01\text{K}$ ergeben sich bereits Unsicherheiten im einstelligen Nanometerbereich. Weiterhin sind auch Verkipungen der Sensoren S1 und S2 besonders kritisch, da dadurch ein Abbe-Fehler erster Ordnung entsteht. Dieser besitzt einen sehr hohen Sensitivitätsfaktor und kann Unsicherheiten von bis zu 100 nm verursachen. Beide kritischen Einflussfaktoren können bei der Konstruktion berücksichtigt werden. Falls die konstruktiven Maßnahmen nicht ausreichen die Unsicherheiten zu minimieren besteht die Möglichkeit einer Korrektur der Messwerte durch Messen der Einflussfaktoren.

6 Ausblick

Vor der Durchführung der Messungen und dem Aufbau der Prototypen wird ein Toleranzmodell der Wechselschnittstelle aufgebaut. Abweichend von der idealen Betrachtung treten Form- und Lageabweichungen von idealen Geometrien aufgrund von Fertigungstoleranzen auf. Zur Beschreibung der Auswirkung dieser Toleranzen wird ein Unsicherheitsmodell erstellt. Damit können die Fehlergrenzen mittels einer Berechnung basierend auf dem Leitfaden zur Angabe der Unsicherheit beim Messen (GUM) bestimmt werden. Die erzielten Ergebnisse und Sensitivitäten fließen in die Konstruktion der Prototypen ein.

Anschließend werden die aufgebauten Prototypen mittels des entwickelten Messaufbaus untersucht. Bei den Prototypen werden sowohl die Materialien der Koppelstellen als auch die Anordnung der Koppelstellen variiert. Beim Einsatz im Vakuum haben insbesondere die Reibverhältnisse und der damit verbundene Einsatz von geeigneten Schmiermitteln einen erheblichen Einfluss auf die Reproduzierbarkeit. Diese wird durch auftretende Stick-Slip Effekte negativ beeinflusst. Eine Alternative Möglichkeit besteht darin gezielt hochfrequente Schwingungen zur Überwindung der

Stick-Slip Effekte einzubringen.

Anhand der erzielten Messergebnisse werden Konstruktionsrichtlinien für die Gestaltung einer Wechselschnittstelle für die NPMM-200 abgeleitet. Der Zusammenhang zwischen Reproduzierbarkeit und Konstruktionsparametern soll in einem Modell basierend auf den Messergebnissen abgebildet werden.

Für den späteren Einsatz in der NPMM-200 ist eine geeignete Einrichtung zum hochreproduzierbaren Aufbringen einer Vorspannkraft zu entwickeln und mit der Wechselschnittstelle zu kombinieren. Die Wechselschnittstelle wird in der NPMM-200 prinzipbedingt entgegen der Gewichtskraft angeordnet. Um die hohe Reproduzierbarkeit der Position der Wechselschnittstelle nicht negativ zu beeinflussen, muss auch das Aufbringen der Vorspannkraft hochreproduzierbar und zeitlich konstant erfolgen.

7 Zusammenfassung

Kinematische Kopplungen zeigen im Allgemeinen eine gute Eignung für ein Toolwechselsystem für die NPMM-200. Diese zeichnen sich in der idealisierten Betrachtung durch eine eindeutige und zwangsfreie Lagedefinition aus. Bei der Gestaltung bieten sich alternative Anordnungsmöglichkeiten, mit denen eine Justierung in drei voneinander unabhängigen Freiheiten möglich ist. Anhand von analytischen Berechnungen wurden grundlegende Eigenschaften ermittelt. Analytische und numerische Modelle bilden die Reproduzierbarkeit nicht in der geforderten Genauigkeit ab. Daher erfolgt die Entwicklung eines Messaufbaus zur hochgenauen Bestimmung der Reproduzierbarkeit der Toolwechselschnittstelle.

8 Literatur

- [1] Balzer, F.: *Entwicklung und Untersuchungen zur 3-D-Nanopositioniertechnik in großen Bewegungsbereichen*. Dissertation (2015). Technische Universität Ilmenau, Ilmenau.
- [2] Füßl, R.; Grünwald, R.; Schmidt, I.: *Messunsicherheitsanalyse von Nanopositionier- und Nanomessmaschinen mit Hilfe eines neuen vektoriellen Modellansatzes*. Technisches Messen: tm: Plattform für Methoden, Systeme und Anwendungen der Messtechnik. (2006) 73 (9), S. 465-471.
- [3] Grimske, S.: *Multifunktionale Schnittstellen für kleine modulare Werkzeugmaschinen*. Dissertation (2014). Helmut-Schmidt-Universität, Hamburg.
- [4] Manske, E.; Jäger, G.; Hausotte, T.: *A multi-sensor approach for complex and large-area applications in micro and nanometrology*. Measure: the journal of measurement science. (2012) 7 (2), S. 44-50.
- [5] Slocum, A.H.: *Kinematic couplings for precision fixturing - Part I. Formulation of design parameters*. Precision Engineering. (1988) 10 (2), S. 85-91.
- [6] Slocum, A. H.; Donmez, A.: *Kinematic couplings for precision fixturing - Part 2. Experimental determination of repeatability and stiffness*. Precision Engineering. (1988) 10 (3), S. 115-122.

Querdynamikregler für Sportwagen

Lateral dynamics controller for sportscars

M. Sc. Yannik Peters, Porsche AG, Porschestraße 911, 71287 Weissach, yannik.peters1@porsche.de

Kurzfassung

Dieser Beitrag stellt eine modulare Stellgrößenarbitrierungseinheit (Control Allocation) eines Querdynamikreglers für Sportwagen vor, die die Querverteilung des Antriebsmoments an der Hinterachse, die Längsverteilung, die Wankmomentverteilung und Hinterachslenkung kombiniert. Die Control Allocation konzentriert sich darauf, die erforderlichen Querkräfte an jeder Achse bereitzustellen und gleichzeitig die Stellerauslenkungen der Aktoren zu minimieren. Darüber hinaus reduziert der verwendete Algorithmus zur Längs- und Querverteilung des Antriebsmoments den erforderlichen Reibwert. Ein Sollgrößengenerator verwendet zwei separate Einzelspurmodelle, um eine gewünschte Gierbeschleunigung und Schwimmwinkelgeschwindigkeit zu berechnen. Ziel ist ein natürliches und vorhersehbares Fahrzeugverhalten, das für ein angenehmes Fahrerlebnis unerlässlich ist. Die gewünschte Fahrzeugbewegung wird von der Control Allocation in die erforderlichen Querkräfte an den Rädern übersetzt. Dieser lateralkraftbasierte Ansatz ermöglicht eine modulare Erweiterbarkeit und Benutzerfreundlichkeit über mehrere Fahrzeugderivate hinweg.

Die vorgeschlagene Steuerung zeigt, dass es möglich ist, Änderungen in der Leistungsfähigkeit von Systemen ohne zeitaufwändiges Tuning auszugleichen und dass die Vernetzung der verfügbaren Systeme dazu beiträgt, die Fahrdynamik und gleichzeitig die Fahrzeugstabilität zu erhöhen.

Abstract

This paper presents a modular control allocation of a lateral dynamics controller for sportscars, which combines lateral torque distribution at the rear axle, longitudinal torque distribution, roll torque distribution and rear wheel steering. The control allocation unit focuses on delivering required lateral forces at each axle, while minimizing actor effort. Moreover the used torque distribution algorithm reduces the required friction value. A reference generator uses two separate single track models in order to calculate a desirable yaw acceleration and side slip angle velocity. The goal is a natural and predictable vehicle behavior, which is essential for a pleasant driving experience. The desired vehicle motion is transferred into required lateral forces at the wheels by the control allocation unit. This lateral force based approach enables modular expandability and usability across multiple vehicles. The proposed controller shows that it is possible to partly compensate for changes in the system's effectiveness without time consuming tuning and that using the available systems in conjunction helps to increase driving performance and the vehicles stability at the same time.

1 Einleitung

Moderne Fahrzeuge verfügen über mehrere Regelsysteme, die die Fahrdynamik des Fahrzeugs maßgeblich beeinflussen. Eines der ersten Systeme war die elektronische Stabilitätskontrolle (ESP), die nach dem Umkippen der A-Klasse von Mercedes beim Elchtest weitgehend eingeführt wurde. Seitdem wurden viele verschiedene Systeme von allen Herstellern entwickelt und eingeführt.

Die steigenden Anforderungen an den Fahrkomfort haben zu einem Anstieg des Fahrzeuggewichts geführt. Um diese erhöhte Masse auszugleichen und der Forderung nach Verbesserungen der Fahrdynamik nach zu kommen, sind Leistung und Drehmoment der Motoren in den letzten Jahrzehnten deutlich gestiegen. Der Allradantrieb wurde verwendet, um Traktionsverluste zu vermeiden, wenn das erhöhte Drehmoment auf die Straße übertragen wird. Der Einsatz beider Achsen erhöht die maximal zulässigen Längskräfte und verbessert damit die Längsdynamik.

Darüber hinaus ermöglichen die Abhängigkeiten zwischen Längs- und Querreifenkräften die Nutzung der Längsverteilung zur gezielten Beeinflussung der Gierbewegung des Fahrzeugs [3], [4].

Systeme zur Variation der Querverteilung des Antriebsmoments, das so genannte Torque Vectoring, nimmt seit einigen Jahren zu und stellt eine ebenfalls effektive Möglichkeit zur Beeinflussung der Gierbewegung dar. Die Einflussmöglichkeiten der Torque Vectoring Systeme hängen oftmals stark von der Fahrsituation und dem Fahrzeugzustand ab [2]. Moderne Fahrzeuge mit Verbrennungsmotor sind beispielsweise mit Sperrdifferentialen oder aktiven Differentialen zur Änderung der Querverteilung ausgestattet. Diese Systeme sind nur in der Lage, das Drehmoment vom schnell drehenden Rad auf das langsamere Rad zu verlagern. Da dies meist das innere Rad ist, lässt sich in vielen Situationen nur ein stabilisierendes Giermoment realisieren. Im Gegensatz dazu sind Überlagerungsdifferenziale in der Lage, nahezu jedes Differenzmoment zwischen den Rädern an einer Achse zu liefern [2].

Im Gegensatz zu Torque Vectoring, das durch asymmetrische Längskräfte direkt ein Giermoment verursacht, beeinflussen Systeme, die die Wankmomentverteilung zwischen

Vorder- und Hinterachse verändern, nur indirekt die Gierbewegung [1]. Die Variation der Wankmomentverteilung führt zu einer Veränderung der dynamischen Radlasten an der Vorder- und Hinterachse und damit zu einer Variation der übertragbaren Querkraften einer Achse. Die Erhöhung des Wankmoments an der Vorderseite führt zu einer Verringerung der Querkraften an der Vorderseite und damit zu einem tendenziell untersteuernden Verhalten.

Die Einführung von Hinterachslenkungen (engl. Rear wheel steering, RWS) ermöglichte eine Verbesserung der Stabilität und Agilität des Fahrzeugs durch gleichsinniges oder gegensinniges Lenken der Hinterräder. Seit ihrem ersten Einsatz in der Nissan Skyline im Jahr 1985 [5] wurden die Systeme drastisch verbessert und sind heute in einer Vielzahl von Fahrzeugen erhältlich.

Der heutige Regelsystemverbund, der in den meisten Fahrzeugen zu finden ist, ist das Ergebnis der evolutionären Entwicklung und Integration neuer Systeme. Die meisten Systeme wurden unabhängig voneinander von verschiedenen Zulieferern entwickelt und erfordern einen zeitaufwendigen Abstimmprozess während der Fahrzeugentwicklung [1]. Daher wurden neue Systeme in den bestehenden Steuerungsverbund integriert, ohne auf mögliche Synergien zu achten. Lediglich negative Wechselwirkungen sollten vermieden werden.

Das Ergebnis ist ein überaktuiertes Fahrzeug, dessen Systeme in einer so genannten friedlichen Koexistenz zusammenarbeiten. Hierbei werden nicht nur negative Interaktionen verhindert, sondern auch die effektive Nutzung von Synergien zwischen den verfügbaren Systemen [1].

Dies hat zu einer Intensivierung der Forschungsarbeiten im Bereich der integrierten dynamischen Regelungen geführt, die einen ganzheitlichen Ansatz im Regeldesign verfolgen, um das ungenutzte Potenzial moderner Fahrzeuge zu nutzen.

Die Überaktuierung moderner Fahrzeuge führt zur Frage der Stellgrößenarbitrierung (engl. Control Allocation), ein Thema, das in vielen Veröffentlichungen in der Luft- und Schifffahrt bereits diskutiert wurde, z.B.[6],[7],[8],[9].

In diesem Beitrag wird ein Fahrzeug, ähnlich dem Porsche 911 Carrera 4 GTS (Typ 991 II) mit Hinterradlenkung, Allradantrieb, Torque Vectoring und variabler Wankmomentverteilung (engl. Roll Torque Distribution, RTD) untersucht. Diese vier Systeme führen zu mehr Eingangsgrößen, als es Variablen gibt, die für die Querdynamik zu regeln sind.

Die in diesem Beitrag vorgestellte Control Allocation nutzt den verbleibenden Freiheitsgrad zur Minimierung des Stellaufwands und liefert gleichzeitig die erforderlichen Querkraften, die von den einzelnen Systemen an den Rädern aufzubringen sind.

2 Modelle

Die Control Allocation verwendet ein erweitertes, nichtlineares Einspurmodell, das zwei Kennlinien für den Lenkwinkel der einzelnen Vorderräder und ein Wankmodell zur

Ermittlung der Radlasten, sowie ein semiempirisches Reifenmodell nach der Similarity Method von Hans Pacejka [10] verwendet.

2.1 Reifenmodell

Die Bewegung des Fahrzeugs ist vollständig abhängig von der Wechselwirkung zwischen den Reifen und der Fahrbahnoberfläche [11]. Um die Auswirkungen von Allradantrieb, Hinterradlenkung, Torque Vectoring und einer variablen Wankmomentverteilung modellieren zu können, muss das Reifenmodell Längsschlupf, Schlupfwinkel und Radlast berücksichtigen.

Nach dem semi-empirischen Modell von Hans Pacejka können die Reifenkräfte wie in den folgenden Gleichungen [12] dargestellt werden:

$$y = D \sin \left[C \tan^{-1} \left((1 - E) B X + E \tan^{-1}(B X) \right) \right] \quad (1)$$

$$Y(x) = y(x) + S_F \quad x = X + S_R \quad (2)$$

Mit den folgenden Ausdrücken:

Tabelle 1 Parameter der Magic Formula

Parameter	Beschreibung
B	Steifigkeitsfaktor
C	Formfaktor
D	Spitzenwertfaktor
E	Krümmungsfaktor
BCD	Schräglaufsteifigkeit
S_F	Vertikaler Offset
S_R	Horizontaler Offset
X	Modeleingang
Y	Modelausgang

Komplexe Berechnungen und zeitaufwändige Parameteridentifikationen sind notwendig, um das Zusammenspiel von Längs- und Querkraft in der vollständigen Magic Formula zu berücksichtigen. Um letzteres zu vereinfachen und die praktische Anwendung zu verbessern, wurde ein Normierungsverfahren eingeführt, ähnlich des in [4] verwendeten.

Dabei werden Schräglaufsteifigkeit und Maximalwert normiert, was zu folgende Ausdruck für die Reifenkräfte führt:

$$\bar{F}(\bar{\sigma}) = \sin \left[\frac{1}{B} \arctan \left[(1 - E) B \bar{\sigma} + E \arctan(B \bar{\sigma}) \right] \right] \quad (3)$$

hierbei $\bar{\sigma}$ der normierte Schlupf und $\bar{F}$ die normierte Kraft. Der normierte Schlupfwert $\bar{\sigma}^*$, an dessen Stelle die maximale Kraft übertragen werden kann, und die normierte Kraft bei durchdrehenden Rädern werden zur Berechnung der Faktoren B und E gemäß der folgenden Formeln genutzt:

$$B = \frac{1}{2 - \frac{2}{\pi} \arcsin \bar{F}_\infty}, B \leq 1 \quad (4)$$

$$E = \frac{B\bar{\sigma}^* - \tan\left(\frac{\pi}{2}B\right)}{B\bar{\sigma}^* - \arctan(B\bar{\sigma}^*)}, E \leq 1 \quad (5)$$

Der normierte Schlupf wird wie folgt berechnet [4]:

$$\bar{\sigma} = \sqrt{\bar{\sigma}_x^2 + \bar{\sigma}_y^2} \quad (6)$$

Hierbei ist $\bar{\sigma}_x$ der normierte Schlupf in Längsrichtung und $\bar{\sigma}_y$ ist der normierte Schlupf in Querrichtung, die sich wie folgt bestimmen lassen:

$$\bar{\sigma}_x = \frac{\lambda_c C_x(F_z)}{\lambda_\mu F_z \mu_x(F_z)} \left(1 - \frac{v_x}{\omega r_e}\right) \quad (7)$$

$$\bar{\sigma}_y = \frac{\lambda_c C_y(F_z)}{\lambda_\mu F_z \mu_y(F_z)} \left(\frac{v_y}{\omega r_e}\right) \quad (8)$$

die hier verwendeten Parameter sind in Tabelle 2 erläutert:

Tabelle II Parameter für die Similarity Method

Parameter	Beschreibung
v_x	Längsgeschwindigkeit
ω	Rotationsgeschwindigkeit des Rades
r_e	Effektiver Rollradius
v_y	Quergeschwindigkeit
$\mu_{x/y}$	Maximalreibung
F_z	Radlast
λ_μ	Faktor für Änderungen von $\mu_{x/y}$
λ_c	Faktor für Änderungen der Steifigkeiten

Die Entnormierung erfolgt gemäß Formel (9) und (10):

$$F_x = \lambda_\mu F_z \mu_x(F_z) \frac{\bar{F} \bar{\sigma}_x}{\bar{\sigma}} \quad (9)$$

$$F_y = \lambda_\mu F_z \mu_y(F_z) \frac{\bar{F} \bar{\sigma}_y}{\bar{\sigma}} \quad (10)$$

2.2 Fahrzeugmodell

Die Control Allocation verwendet Hinterradlenkung, Torque Vectoring, Allradantrieb und eine variable Wankmomentverteilung. Ein einfaches einspuriges Modell ist daher nicht ausreichend. Ein Modell, das asymmetrische Längskräfte an einer Achse zulässt und verschiedene Radlasten berücksichtigen kann, ist erforderlich. Um die Berechnungen so einfach wie möglich zu halten und den Rechenaufwand zu reduzieren, werden die Reifenkräfte an jedem Rad einzeln berechnet und dann für jede Achse addiert. Somit können die Bewegungsgleichungen des einspurigen Modells genutzt werden:

$$\ddot{\psi} \cdot J_{zz} = l_F \cdot F_{yF} - l_R \cdot F_{yR} + M_z^{act} \quad (11)$$

$$\dot{\beta} = \frac{F_{yF} + F_{yR}}{mv} - \dot{\psi} \quad (12)$$

dabei ist $\ddot{\psi}$ die Gierbeschleunigung, $\dot{\beta}$ die Schwimmwinkelgeschwindigkeit, J_{zz} das Trägheitsmoment um die Hochachse, m die Fahrzeugmasse und $l_{F/R}$ die Abstände der Achsen zum Schwerpunkt. $F_{yF/yR}$ sind die addierten Querkräfte an Vorder- beziehungsweise Hinterachse und M_z^{act} beschreibt das sich aus Torque Vectoring ergebende Giermoment.

Letzteres wird wie folgt aus den asymmetrischen Längskräften F_{xRR} und F_{xRL} an der Hinterachse berechnet:

$$M_z^{act} = (F_{xRR} - F_{xRL}) \cdot \frac{s_R}{2} \quad (13)$$

s_R steht hierbei für die Spurweite an der Hinterachse.

Zur Regelung der Wankmomentverteilung ist ein Wankmodell des Aufbaus zwingend erforderlich. Die Wankbewegung des Aufbaus lässt sich mit der folgenden Differentialgleichung beschreiben:

$$J_{xx} \cdot \ddot{\phi} = m_{Body} \cdot a_y - c_{stiff} \cdot \phi - d_{damp} \dot{\phi} \quad (14)$$

Hier ist J_{xx} das Trägheitsmoment um die Fahrzeuglängsachse, m_{Body} die Aufbaumasse, a_y die Querbeschleunigung, c_{stiff} die Rollsteifigkeit der Spiralfedern und Stabilisatoren und d_{damp} die Rolldämpfung aller vier Dämpfer zusammengenommen.

Die Einflüsse von Aerodynamik, Wankwinkel, Längsbeschleunigung und Querbeschleunigung fließen wie folgt in die Radlastberechnung der einzelnen Räder:

$$F_{z,FL} = \frac{F_{z,F,aero}}{2} - F_{z,lgt} - F_{z,lat} + \frac{F_{z,F,stat}}{2} \quad (15)$$

$$F_{z,FR} = \frac{F_{z,F,aero}}{2} - F_{z,lgt} + F_{z,lat} + \frac{F_{z,F,stat}}{2} \quad (16)$$

$$F_{z,FL} = \frac{F_{z,R,aero}}{2} + F_{z,lgt} - F_{z,lat} + \frac{F_{z,R,stat}}{2} \quad (17)$$

$$F_{z,FL} = \frac{F_{z,R,aero}}{2} + F_{z,lgt} + F_{z,lat} + \frac{F_{z,R,stat}}{2} \quad (18)$$

Die Auf- beziehungsweise Abtriebskräfte an Vorder- und Hinterachse lassen sich aus der Luftdichte ρ , der Stirnfläche A des Fahrzeuges, den Auftriebsbeiwerten für Vorder- und Hinterachse $c_{aF/R}$ sowie der Fahrgeschwindigkeit v berechnen:

$$F_{z,F,aero} = \frac{1}{2} \rho A c_{aF} v^2 \quad (19)$$

$$F_{z,R,aero} = \frac{1}{2} \rho A c_{aR} v^2 \quad (20)$$

Radlaständerungen durch die Längsbeschleunigung a_x errechnen sich aus dem Zusammenhang:

$$F_{z,igt} = \frac{1}{2} \frac{h_{CoG} m a_x}{l} \quad (21)$$

hierbei sind h_{CoG} die Schwerpunkthöhe, l der Radstand, und s_F respektive s_R die Spurweiten vorne und hinten.

Die Wankmomentverteilung c_{WA} beeinflusst die Radlastverlagerungen unter Querschleunigung. Die Gesamtrollsteifigkeit des Fahrzeuges wird mittels der Wankmomentverteilung anteilig auf der Vorder- und Hinterachse verteilt:

$$c_{Stiff,F} = c_{Stiff} \cdot c_{WA} \quad (22)$$

$$c_{Stiff,R} = c_{Stiff} \cdot (1 - c_{WA}) \quad (23)$$

Hieraus lassen sich dann die Radlaständerungen $F_{z,F,lat}$ und $F_{z,R,lat}$, die aus der Querschleunigung resultieren, berechnen:

$$F_{z,F,lat} = \frac{\varphi \cdot c_{Stiff,F} + \dot{\varphi} \cdot d_{damp,F}}{s_F} \quad (24)$$

$$F_{z,R,lat} = \frac{\varphi \cdot c_{Stiff,R} + \dot{\varphi} \cdot d_{damp,R}}{s_R} \quad (25)$$

Dabei sind $d_{damp,F/R}$ die Wankdämpfung an Vorder- und Hinterachse.

Für die Schräglaufwinkel beim Einspurmodell gelten gemäß [13] die folgenden Zusammenhänge:

$$\alpha_F = \delta_F - \frac{\dot{\psi} l_F}{v} - \beta \quad (26)$$

$$\alpha_R = \delta_R + \frac{\dot{\psi} l_R}{v} - \beta \quad (27)$$

Dabei beschreiben $\delta_{F/R}$ die Lenkwinkel an den Achsen. Da der Längsschlupf einen Eingang der Control Allocation darstellt, muss dieser innerhalb des Models nicht berechnet werden.

3 Sollgrößengenerator

Ziel der vorgeschlagenen Vorsteuerung und der Control Allocation ist ein natürliches, reproduzierbares und zuverlässiges Fahrzeugverhalten in allen Fahrsituationen, ähnlich dem Ziel von [13]. Nur so kann der Fahrer das volle Potenzial seines Autos ausschöpfen.

Heute führt die friedliche Koexistenz von Regelsystemen manchmal dazu, dass verschiedene Akteure das Fahrzeugverhalten gegenläufig beeinflussen, was zu einem unvorhersehbaren Fahrverhalten führt [13] und somit das Vertrauen des Fahrers in sein Fahrzeug verringert.

Um dieses synthetische Fahrgefühl zu vermeiden, das oft durch geschlossene Regelkreise verursacht wird, kommt hier eine Vorsteuerung zum Einsatz.

Diese verwendet zwei Einspurmodelle: zum Einen das Einspurmodell eines sportlichen, passiven Fahrzeugs als Referenz und als Zweites das Einspurmodell des aktiven Fahrzeugs, in dem die Steuerung tatsächlich implementiert ist.

Das Referenzfahrzeug ist durch Variation von Gewicht, Radstand, Aerodynamik, Spurweite und Gierträgheitsmoment abstimmbar.

Die Querfahrtdynamik des Fahrzeugs wird durch die folgenden Gleichungen beschrieben:

$$\ddot{\psi} \cdot J_{zz}^{ref} = l_F^{ref} \cdot F_{yF}^{ref} - l_R^{ref} \cdot F_{yR}^{ref} \quad (28)$$

$$\dot{\beta}^{ref} = \frac{F_{yF}^{ref} + F_{yR}^{ref}}{m^{ref} v} - \dot{\psi}^{ref} \quad (29)$$

Die aktiven Regelsysteme werden teilweise direkt und teilweise indirekt modelliert, indem die Berechnung der Querkraften beeinflusst wird.

Zur Abbildung des Einflusses des Torque Vectoring wird das zweite Einspurmodell um ein Giermoment M_z^{act} erweitert. Hinterradlenkung und Wankmomentverteilung beeinflussen die Berechnung der Querkraften und müssen daher nicht als separate Terme implementiert werden.

$$\ddot{\psi} \cdot J_{zz} = l_F \cdot F_{yF}^{act} - l_R \cdot F_{yR}^{act} + M_z^{act} \quad (30)$$

$$\dot{\beta} = \frac{F_{yF}^{act} + F_{yR}^{act}}{m v} - \dot{\psi} \quad (31)$$

Um die Querkraften an jedem Rad für beide Fahrzeugmodelle zu berechnen, werden die gemessene Geschwindigkeit des Fahrzeugs, die Längsschleunigung, der Lenkwinkel und der Längsschlupf verwendet. Aus dem Lenkwinkel ergeben sich gemäß zweier Kennlinien die Lenkwinkel der beiden Vorderräder, sodass auch die Lenkgeometrie korrekt abgebildet wird. Die Querkraften jeder Achse werden addiert.

Die Querkraften, die erforderlich sind, um die gleiche Fahrzeugreaktion im aktuellen Zustand für das passive und aktive Fahrzeug zu erreichen, lassen sich aus den folgenden Forderungen ermitteln:

$$\ddot{\psi}^{ref} = \ddot{\psi} \quad (32)$$

$$\dot{\beta}^{ref} = \dot{\beta} \quad (33)$$

Aus den Zusammenhängen (28), (29), (30) und (31) folgt:

$$\begin{aligned} & \frac{l_F \cdot F_{yF}^{act} - l_R \cdot F_{yR}^{act} + M_z^{act}}{J_{zz}} \\ &= \frac{l_F^{ref} \cdot F_{yF}^{ref} - l_R^{ref} \cdot F_{yR}^{ref}}{J_{zz}^{ref}} \end{aligned} \quad (34)$$

$$\frac{F_{yF}^{act} + F_{yR}^{act}}{m v} - \dot{\psi} = \frac{F_{yF}^{ref} + F_{yR}^{ref}}{m^{ref} v} - \dot{\psi}^{ref} \quad (35)$$

Um die erforderlichen Querkraften zu erhalten, die notwendig sind damit das aktive Fahrzeug ähnlich wie das passive Fahrzeug im gegebenen Zustand reagiert, müssen die Gieraten und damit der Ausgangszustand identisch sein. Nach

Lösen der Gleichungen (34) und (35) stellen sich die Querkräfte wie folgt dar:

$$F_{yF}^{act} = \frac{(-M_z^{act} \cdot m^{ref} + m \cdot l_R \cdot (F_{yF}^{ref} + F_{yR}^{ref}))}{m^{ref} \cdot (l_F + l_R)} \quad (36)$$

$$+ \frac{J_{zz} \cdot (F_{yF}^{ref} \cdot l_F^{ref} - F_{yR}^{ref} \cdot l_R^{ref})}{J_{zz}^{ref} \cdot m^{ref} \cdot (l_F + l_R)}$$

$$F_{yR}^{act} = \frac{(M_z^{act} \cdot m^{ref} + m \cdot l_F \cdot (F_{yF}^{ref} + F_{yR}^{ref}))}{m^{ref} \cdot (l_F + l_R)} \quad (37)$$

$$- \frac{J_{zz} \cdot (F_{yF}^{ref} \cdot l_F^{ref} - F_{yR}^{ref} \cdot l_R^{ref})}{J_{zz}^{ref} \cdot m^{ref} \cdot (l_F + l_R)}$$

Die Querkräfte, die von den Systemen aktiv gestellt werden müssen, damit das aktive Fahrzeug die selbe Zustandsänderung erfährt wie das Referenzfahrzeug, werden in der Control Allocation in Abschnitt 4 berechnet.

Der vorgestellte Referenzgenerator führt zu ähnlichen Ergebnissen wie die dem Stand der Technik entsprechenden Kennfelder, die die Eingriffe der Systeme basierend auf dem Zustand des Fahrzeugs bestimmen. Im Gegensatz zu Kennfeldern, die intensive und zeitaufwändige Tests und iterative Optimierungen erfordern, ist dieser modellbasierte Ansatz durch die Änderung der Fahrzeugdaten leicht abstimmbaar.

Während ähnliche Arbeiten eine Minimierung des Schwimmwinkels anstreben [15], führt das vorgeschlagene Verfahren zu einem Verhalten eines Referenzfahrzeugs in Übereinstimmung mit V. Berkefelds Untersuchung über die subjektive Bewertung einer Kompensation des Schwimmwinkels [14].

4 Control Allocation

Die Control Allocation ist in der Lage, den Hinterradlenkwinkel, die Längs- und Querdrehmomentverteilung an der Hinterachse und die Wankmomentverteilung zu steuern. Abschnitt 4.1 erläutert die obere Ebene der Stellgrößenarbitrierung, bevor Abschnitt 4.2 das Verfahren zur Bestimmung der Antriebsmomentverteilung behandelt.

4.1 Obere Ebene der Control Allocation

Durch Subtrahieren der Querkräfte des aktiven Fahrzeugs von den notwendigen Kräften, die in Abschnitt III berechnet wurden, erhalten wir die Kräfte, die erforderlich sind, um das Verhalten des passiven Referenzfahrzeugs zu erreichen:

$$F_{yF}^{req} = F_{yF}^{act} - F_{yF} \quad (38)$$

$$F_{yR}^{req} = F_{yR}^{act} - F_{yR} \quad (39)$$

$F_{yF/R}$ bezeichnet die Seitenkräfte am Einspurmodell des aktiven Fahrzeugs. Das Giermoment ergibt sich zu:

$$M_z^{req} = F_{yF}^{req} \cdot l_F - F_{yR}^{req} \cdot l_R \quad (40)$$

Hauptziel der Control Allocation ist die Realisierung der erforderlichen aktiven Kräfte F_{yF} und F_{yR} . Die Hinterradlenkung beeinflusst ausschließlich die Querkraft an der Hinterachse, während alle anderen verfügbaren Systeme beide Querkräfte gleichzeitig beeinflussen. Die Überaktuierung ermöglicht die Einführung von Nebenzielen. In diesem Fall werden die Stellerauslenkungen minimiert. Dies führt zu folgendem Optimierungsproblem, bei dem das Kriterium Q minimiert werden soll:

$$Q = \frac{1}{2} u_{tot}^T \cdot u_{tot} + \lambda^T \cdot (F_{req} - B \cdot u) \quad (41)$$

F_{req} ist dabei ein Vektor, der die beiden angeforderten Seitenkräfte beinhaltet, B ist die Effektivitätsmatrix der Systeme, u ist ein Vektor, der die auf ihren jeweiligen Stellbereich normierten Stelleränderungen beinhaltet, u_{tot} beschreibt die analog normierten Stellerauslenkungen und Stelleränderungen, und λ ist ein Vektor, der die beiden die Lagrange Multiplikatoren beinhaltet.

Aufgrund des engen Zusammenhangs zwischen Quer- und Längsdrehmomentverteilung sind diese Systeme in einer unteren Ebene der Stellgrößenarbitrierungseinheit zusammengefasst, die als aktive Giermomentgeber dienen. Die obere Ebene der Stellgrößenarbitrierungseinheit fordert von diesem aktiven Giermomentgeber ein Giermoment an, das dann die Verteilung berechnet, die nicht nur die Drehmomentanforderung erfüllt, sondern auch den erforderlichen maximalen Reibungswert minimiert. Die Berechnungen werden in Abschnitt 4.2 erläutert.

Die maximale Effektivität der Antriebsmomentverteilung kann über den folgenden Ausdruck vereinfacht dargestellt werden:

$$B_{torque} = \begin{bmatrix} \frac{M_z^{max}}{l_F + l_R} \\ -\frac{M_z^{max}}{l_F + l_R} \end{bmatrix} \quad (42)$$

Der Eintrag für die Wankmomentverteilung in der Effektivitätsmatrix ergibt sich aus der funktionalen Abhängigkeit der Seitenkraft von der Radlast. Mittels des Reifenmodells aus Abschnitt 2.1 werden die Gradienten $\frac{\partial F_y}{\partial F_z}$ der Seitenkraft über der Radlast berechnet. Anschließend lässt sich der Eintrag für die Effektivitätsmatrix wie folgt bestimmen:

$$B_{roll} = \begin{bmatrix} \frac{M_x \cdot \left(\frac{\partial F_{yFR}}{\partial F_{zFR}} - \frac{\partial F_{yFL}}{\partial F_{zFL}} \right) (c_{WA}^{max} - c_{WA}^{min})}{2S_F} \\ -\frac{M_x \cdot \left(\frac{\partial F_{yRR}}{\partial F_{zRR}} - \frac{\partial F_{yRL}}{\partial F_{zRL}} \right) (c_{WA}^{max} - c_{WA}^{min})}{2S_R} \end{bmatrix} \quad (43)$$

Mit einem ähnlichen Vorgehen lässt sich auf die Hinterachslenkung implementieren. Hierbei wird der aktuelle Gradient $\frac{\partial F_{yR}}{\partial \alpha_R}$ der Seitenkraft über dem Schräglaufwinkel an der Hinterachse benötigt:

$$B_{RWS} = \begin{bmatrix} 0 \\ \left(\frac{\partial F_{yRL}}{\partial \alpha_R} + \frac{\partial F_{yRR}}{\partial \alpha_R} \right) \delta_R^{max} \end{bmatrix} \quad (44)$$

In Kombination von (42), (43) und (44) ergibt sich die folgende Effektivitätsmatrix:

$$B = [B_{torque} \quad B_{roll} \quad B_{RWS}] \quad (45)$$

Der normierte Vektor der Stelleränderungen lautet:

$$u = \begin{bmatrix} \frac{M_z}{M_z^{max}} \\ \frac{2c_{WA}}{(c_{WA}^{max} - c_{WA}^{min})} \\ \frac{\delta_R}{\delta_R^{max}} \end{bmatrix} \quad (46)$$

Und der zu minimierende Vektor der normierten Stellerauslenkungen und Stelleränderungen wird beschrieben durch:

$$u_{tot} = \begin{bmatrix} \frac{M_z}{M_z^{max}} \\ \frac{2c_{WA}}{(c_{WA}^{max} - c_{WA}^{min})} \\ \frac{\delta_R}{\delta_R^{max}} \\ \frac{M_z + M_z^{cur}}{M_z^{max}} \\ \frac{2(c_{WA} + c_{WA}^{cur})}{(c_{WA}^{max} - c_{WA}^{min})} \\ \frac{\delta_R + \delta_R^{cur}}{\delta_R^{max}} \end{bmatrix} \quad (47)$$

Hierbei steht der Index „cur“ für die Stellerauslenkungen im vorherigen Zeitschritt. Das Lagrange Multiplikatorverfahren führt zum folgenden Gleichungssystem:

$$\frac{u_1 + u_1^{cur}}{u_1^{max}} - u_1^{max}(\lambda_1 b_{11} + \lambda_2 b_{21}) + u_1 = 0 \quad (48)$$

$$\frac{u_2 + u_2^{cur}}{u_2^{max}} - u_2^{max}(\lambda_1 b_{12} + \lambda_2 b_{22}) + u_2 = 0 \quad (49)$$

$$\frac{u_3 + u_3^{cur}}{u_3^{max}} - u_3^{max}(\lambda_1 b_{11} + \lambda_2 b_{21}) + u_3 = 0 \quad (50)$$

$$F_{yF}^{req} - \frac{b_{11}u_1}{u_1^{max}} - \frac{b_{12}u_2}{u_2^{max}} - \frac{b_{13}u_3}{u_3^{max}} = 0 \quad (51)$$

$$F_{yR}^{req} - \frac{b_{21}u_1}{u_1^{max}} - \frac{b_{22}u_2}{u_2^{max}} - \frac{b_{23}u_3}{u_3^{max}} = 0 \quad (52)$$

Aus der Lösung dieser Gleichungen resultieren die Stelleränderungen, die erforderlich sind um die Querkräfte bereitzustellen.

Bisher berücksichtigt die vorgeschlagene Control Allocation die Einschränkungen der Aktoren nicht. Die Berücksichtigung solcher Einschränkungen kann dazu führen, dass der Referenzgenerator nicht realisierbare Kräfte, anfordert, und damit keine zielführende Lösung möglich ist. Um dieses Problem zu lösen, werden die erforderlichen Kräfte vorab begrenzt und die Aktoren nacheinander saturiert. Zunächst werden die maximalen und minimalen Querkräfte, die von den aktiven Systemen erzeugt werden können, aus der Effektivitätsmatrix und den Minimal- und Maximalwerten für den Eingangsvektor u berechnet. Um sicherzustellen, dass bei unerreichbaren Sollkräften mindestens das erforderliche Giermoment erreicht wird, wird das in Abb. 1 dargestellte Verfahren verwendet.

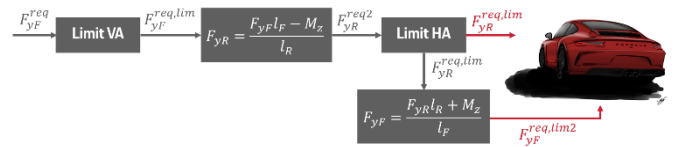


Abb. 1: Verfahren zur Limitierung der Seitenkrafteanforderung unter Beibehaltung des Giermoments

Als Erstes wird die erforderliche Querkraft an der Vorderachse durch die maximal und minimal mögliche Querkraft an der Vorderachse begrenzt. Anschließend wird die Querkraft an der Hinterachse berechnet, die zur Aufrechterhaltung des gewünschten Giermoments erforderlich ist, bevor sie ebenfalls begrenzt wird. Abschließend wird die für das gewünschte Giermoment erforderliche Querkraft an der Vorderseite berechnet. Nach diesem Verfahren ist sichergestellt, dass Kräfte, die von der Control Allocation angefordert werden, erreichbar sind und zum gewünschten Giermoment führen.

Obwohl die erforderlichen Kräfte immer erreichbar sind, ist es möglich, dass die Control Allocation eine Lösung ermittelt, die einen Aktor über seine Grenzen hinaus beansprucht, ohne einen anderen Akteur in seinem vollen Potenzial zu nutzen. In diesem Fall wird überbeanspruchte System saturiert, seine Effektivität auf Null gesetzt und die verbleibende erforderliche Kraft berechnet. Die neue benötigte Kraft wird dann wieder an die Control Allocation abgegeben. Dieser Vorgang wird so lange wiederholt, bis kein Aktor mehr überbeansprucht wird.

4.2 Untere Ebene der Control Allocation

Die Control Allocation innerhalb der Drehmomentverteilung ist aufgrund der Wechselwirkungen zwischen beiden Systemen komplizierter. Sie erfolgt nachdem zuvor bereits in [16] dargestellten Verfahren.

Beide Systeme, Quer- und Längsverteilung, sind in der Lage, ein Giermoment zu erzeugen. Somit gibt es einen Freiheitsgrad, mit dem der erforderliche Reibwert μ minimiert wird. Der erforderliche Reibwert für jedes Rad ij ist wie folgt definiert:

$$\mu = \frac{\sqrt{F_{x,ij}^2 + F_{y,ij}^2}}{F_{z,ij}} \quad (53)$$

Abb. 2 veranschaulicht das Verfahren zur Minimierung des erforderlichen Reibungswertes unter Beibehaltung des gewünschten Giermoments.

Der erste Schritt ist die Berechnung des anfänglichen Differenzmoments an der Hinterachse:

$$M_{TV}^{init} = \frac{M_z^{act}}{s_R} \cdot \frac{r_{dyn}}{2} \quad (54)$$

Dabei beschreibt r_{dyn} den dynamischen Rollradius und s_R die Spurweite an der Hinterachse. Ausgehend von der aktuellen Längsverteilung und dem resultierenden Differenzmoment der Gleichung (54) werden die Querkräfte an jedem Rad berechnet.

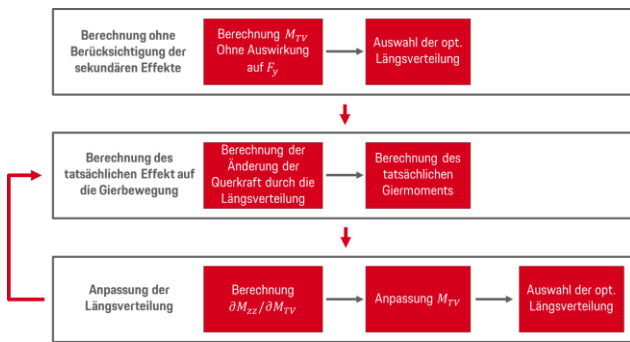


Abb. 2: Verfahren zur Berechnung der Längs- und Querverteilung [16]

Der nächste Schritt ist die Berechnung der optimalen Längsverteilung, die mit einem iterativen Ansatz erfolgt: Die erforderlichen Reibwerte an jedem Rad werden für Längsverteilungen zwischen Null (Vorderradantrieb) und 100% (Hinterradantrieb) mit einer Schrittweite von 5% berechnet. Die Verwendung einer Schrittweite kleiner 5% ist nicht erforderlich, da der Aktor nicht über eine höhere Auflösung verfügt. Abb. 3 zeigt den Einfluss der Längsmomentverteilung auf den erforderlichen Reibwert.

In [16] wurde das sekundäre Giermoment untersucht, das durch die Wechselwirkung zwischen Längs- und Querkräfte an den Rädern verursacht wird. Um diese gegenseitige Beeinflussung auszugleichen, werden die Änderungen der Querkräfte, die Auswirkung auf das Gesamtgiermoment und die Steigung $\partial M_{zz}/\partial M_{TV}$ berechnet. Dadurch kann das Differenzmoment an der Hinterachse eingestellt werden, was sich wiederum auf die gewünschten Reibwerte und damit die optimale Längsverteilung auswirkt. Theoretisch muss dieser Prozess wiederholt werden, bis sich die

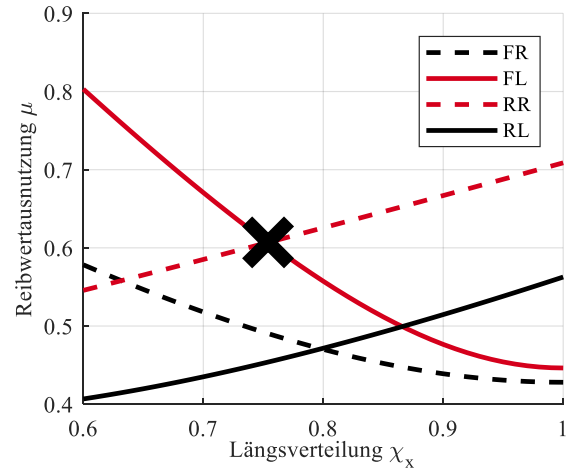


Abb. 2: benötigte Reibwerte in Abhängigkeit der Längsverteilung des Antriebsmoments [16]

Ergebnisse nicht mehr ändern, aber Abb. 4 zeigt, dass die Ergebnisse bereits nach einer Iteration innerhalb der Genauigkeit des Aktors von 5% liegen. Daher sind zusätzliche Iterationen nicht erforderlich und die Ergebnisse der ersten Iteration für das Differenzmoment an der Hinterachse M_{TV} und der Längsverteilung χ_x können verwendet werden.

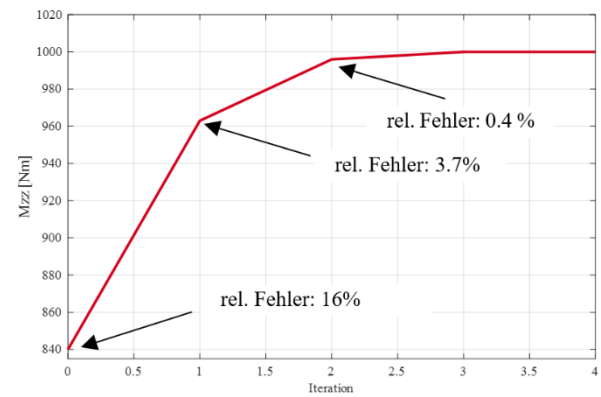


Abb. 3: Ergebnisse mehrere Iterationen des Verfahrens [16]

5 Ergebnisse

Zur Validierung und zum Test der vorgestellten Control Allocation wird die Porsche eigene Simulationsumgebung (SimEnv) verwendet, die ein Fahrzeugmodell mit 14 Freiheitsgraden einschließlich der vier Radrotationen verwendet. Das Modell ist in die folgenden Subsysteme unterteilt:

- Aufbaumodell
 - Aufhängung
 - Lenkung
 - Aerodynamik
- Antriebsstrang
- Bremse
- Reifen
- Fahrbahn
- Fahrer
- Regelsysteme

Das Reifen-Subsystem verwendet die in [10] erläuterte Magic Formula. Innerhalb des Subsystems Aufhängung sind

Nichtlinearitäten der Achskinematik, Elastokinematik sowie Feder- und Dämpfungsverhalten in Kennlinien hinterlegt. Die Daten der Kennlinien werden in der Mehrkörpersimulationssoftware Adams (Automatic Dynamic Analysis of Mechanical Systems) bestimmt.

Zum Testen der Control Allocation werden ein Fahrzeug, ähnlich einem Porsche Carrera 911 4 GTS (Typ 991 II), und ein aktives Fahrzeug mit Allradantrieb (4WD), Hinterradlenkung (RWS), variabler Rollmomentverteilung (RTD) und elektronischem Überlagerungsdifferential, verwendet. Ein Lenkwinkelsprung auf 50° bei 100 km/h wird mit unterschiedlichen Fahrzeugkonfigurationen und unterschiedlichen maximalen Stellmöglichkeiten der Torque Vectoring Einheit simuliert, um die Anpassungsfähigkeit des Algorithmus zu prüfen.

Das Referenzfahrzeug hat einen längeren Radstand (100mm größer) und der Schwerpunkt wird um 50 mm in Richtung Vorderachse verschoben. Außerdem wird die Aerodynamik verbessert und erzeugt etwas mehr Abtrieb an der Vorderachse. Diese Anpassungen führen zu einem agileren und gleichzeitig stabileren Fahrzeugverhalten.

Abb. 5 zeigt die Gierrate für einen Lenkwinkelsprung bis 50° Lenkradwinkel bei einer Geschwindigkeit von 100 km/h für unterschiedliche Torque Vectoring Konfigurationen. Im Vergleich zum passiven Fahrzeug wird die stationäre Gierrate erhöht, während das Überspringen reduziert wird. Auch der in Abb. 6 dargestellte Schwimmwinkel wird deutlich reduziert und damit die Stabilität des Fahrzeugs verbessert. Die Unterschiede zwischen den Fahrzeugkonfigurationen sind sehr gering, obwohl Abb. 7 zeigt, dass je nach Konfiguration die Wankmomentverteilung und die variable Drehmomentverteilung bis zu ihrem maximalen Potenzial genutzt werden. Von besonderem Interesse ist die Konfiguration mit abgeschaltetem Torque Vectoring. In diesem Fall wird sogar die Hinterradlenkung verwendet, um die Agilität des Fahrzeugs zu erhöhen, indem zunächst in die entgegengesetzte Richtung der Vorderräder gelenkt wird, um das Fehlen von Torque Vectoring und die Sättigung von RTD auszugleichen. Danach lenkt RWS schnell in die gleiche Richtung wie die Vorderräder, um den Schwimmwinkel zu verringern und die Stabilität des Fahrzeugs zu gewährleisten.

Das Ergebnis sind nahezu identische Gierraten und Schwimmwinkel für die getesteten Konfigurationen. Es waren keine Anpassungen des Algorithmus oder der Parameter erforderlich, um diese Ergebnisse zu erzielen.

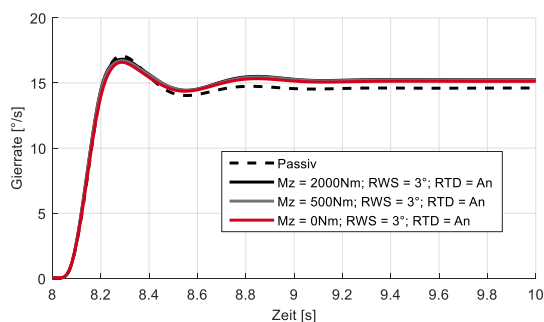


Abb. 4: Gierrate während des Lenkwinkelsprungs

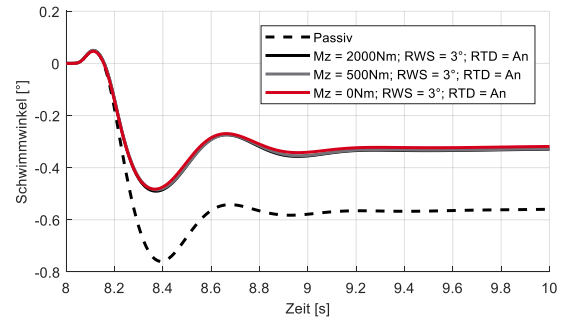


Abb. 5: Schwimmwinkel während des Lenkwinkelsprungs

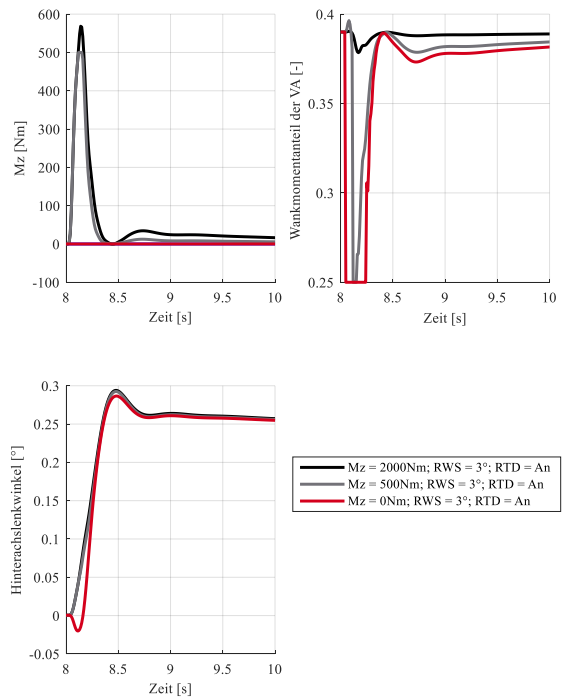


Abb. 6: Stellerauslenkungen während des Lenkwinkelsprungs

6 Fazit

Eine modulare Control Allocation für das Drehmomentmanagement, die variable Wankmomentverteilung und die Hinterradlenkung wurde vorgestellt. Die vorgeschlagene Stellgrößenarbitrierung konzentriert sich auf das Erreichen der erforderlichen Kräfte, die von einem Referenzgenerator berechnet werden. Der Referenzgenerator berechnet aus zwei separaten Einspurmodellen die Kräfte, die erforderlich sind, um das tatsächliche, aktive Fahrzeug in eine verbesserte passive Version von sich selbst zu verwandeln. Die berechneten Kräfte werden dann innerhalb der Stellgrößenarbitrierung begrenzt, um sicherzustellen, dass die angeforderten Kräfte tatsächlich erreichbar sind. Ist dies nicht der Fall, werden die Kräfte entsprechend dem gewünschten Giermoment angepasst, um eine zufriedenstellende Gierbewegung zu gewährleisten.

Die Stellgrößenarbitrierung minimiert die Auslenkungen und Änderungsraten der Aktoren und erreicht dabei die gewünschten Kräfte. Ein Verfahren zur Längs- und Querdrehmomentverteilung stellt sicher, dass das angeforderte Giermoment erfüllt wird und gleichzeitig der erforderliche Reibwert reduziert wird.

Der in Abschnitt II erläuterte Referenzgenerator verfolgt das Ziel eines natürlichen Fahrverhaltens, indem er nicht versucht, ein synthetisches, ideales Fahrverhalten zu erreichen, sondern das Verhalten eines passiven Fahrzeugs. Wie alle modellbasierten Referenzgeneratoren ermöglicht das vorgeschlagene Konzept eine einfache Übertragung auf Fahrzeugderivate.

Simulationsergebnisse in Abschnitt 5 haben gezeigt, dass die Stellgrößenarbitrierung in der Lage ist, Unterschiede in den Kapazitäten oder der Verfügbarkeit des aktiven Systems durch entsprechende Nutzung der übrigen Systeme auszugleichen. Zeitaufwändiges Tuning aller möglichen Fahrzeugkonfigurationen wird dadurch nahezu überflüssig.

Der modulare Aufbau ermöglicht auch die einfache Integration weiterer Systeme. Solange es möglich ist, deren Wirkung auf die Querkraften an Vorder- und Hinterachse zu beschreiben, kann die Stellgrößenarbitrierung sie effektiv nutzen.

7 Literatur

- [1] N. Laumanns, "Integrale Reglerstruktur zur effektiven Abstimmung von Fahrdynamikregelsystemen", Ph.D. dissertation, Automobiltechnik, Institut für Kraftfahrwesen, Aachen, Oct. 2007, ISBN: 978-3-925194-90-0
- [2] T. C. Meißner, "Verbesserung der Fahrzeugquerdynamik durch variable Antriebsmomentverteilung", Ph.D. dissertation, Göttingen: Cuvillier Verlag, 2008, ISBN: 978-3-86727-602-3.
- [3] D. Piyabongkarn, R. Rajamani, J. Y. Lew, and H. Yu, "On the use of torque-biasing devices for vehicle stability control," in Proc. Amer. Control Conf. (ACC), Minneapolis, MN, 2006, pp. 5360–5365.
- [4] M. Struijk, "Central Differential Control of an Active Four Wheel Drive Vehicle", M.S. thesis, Delft Center for Systems and Control, TU Delft, 2015
- [5] Pfeffer, Peter; Harrer, Manfred: „Lenkungs-handbuch“, Wiesbaden: Vieweg+Teubner Verlag, 2011. - ISBN 978-3-834-88167-0.
- [6] X. Shao, Z. Liang, B. Chen and C. Liu, "A modified weighted pseudo-inverse control allocation using genetic algorithm," 2015 34th Chinese Control Conference (CCC), Hangzhou, 2015, pp. 5554-5559. doi: 10.1109/ChiCC.2015.7260507
- [7] Antonelli, G. and Chiaverini, S., 1998, "Task-Priority Redundancy Resolution for Underwater Vehicle-Manipulator Systems," Proceedings of the 1998 IEEE International Conference on Robotics & Automation, pp. 768 – 773
- [8] Bodson, M., 2002, "Evaluation of Optimization Methods for Control Allocation," Journal of Guidance, Control, and Dynamics, Vol. 25, No. 4, pp. 703 – 711
- [9] Bordignon, K. A. and Durham, W. C., 1995, "Closed-Form Solutions to Constrained Control Allocation Problem", Journal of Guidance, Control, and Dynamics, Vol. 18, No. 5, pp. 1000 – 1007.
- [10] Manual, MF-Tyre U.: MF-Tyre User Manual Version 5.2. DELFT-TYRE, TNO, 2002
- [11] A. Bhoraskar, P. Sakthivel, "A Review and a Comparison of Dugoff and Modified Dugoff Formula with Magic Formula" 2017 International Conference on Nascent Technologies in the Engineering Field (IC-NTE-2017), 2017, doi: 10.1109/IC-NTE.2017.7947898
- [12] R. Gnadler, "Fahreigenschaften von Kraftfahrzeugen II", Scriptum, FAST-Institute, KIT, Karlsruhe, Germany, 2016
- [13] König L., Walter T., Gutmayer B., Merlein D. (2014) Integrated Vehicle Dynamics Control – an optimized approach for linking multiple chassis actuators. In: Bargende M., Reuss HC., Wiedemann J. (eds) 14. Internationales Stuttgarter Symposium. Proceedings. Springer Vieweg, Wiesbaden
- [14] V. Berkefeld „Theoretische Untersuchungen zur Vier-radlenkung, Stabilität und Manövrierbarkeit“. HDT Conference T-30-930-056-9, 1989, [Online] Available: <http://www.d3dhemmer.com/Patente+Ver-suche/4WS/Theoretische%20Unter-suchungen%20zur%20Vierradlenkung.pdf>
- [15] Elmarakbi, A., Rengaraj, C., Wheatley, A. et al. Int. J. Dynam. Control (2013) 1: 360. <https://doi.org/10.1007/s40435-013-0026-9>
- [16] Peters, Y. Stadelmayer, M., „Control Allocation for All Wheel Drive sportscars with Rear Wheel Steering“, currently under review at Automotive and Engine Technology

Vergleich von Graphen- und Kalman-Filter-basierten Verfahren zur verbesserten Eigenbewegungsschätzung auf Basis detektierter statischer Objekte

Niklas Stannartz, M.Sc., TU Dortmund, Institut für Roboterforschung, Lehrstuhl für Regelungssystemtechnik, 44221 Dortmund, Deutschland, niklas.stannartz@tu-dortmund.de

Dr.-Ing. Avraam Tolmidis, ZF, Automated Driving & Integral Cognitive Safety, 40547 Düsseldorf, Deutschland, avraam.tolmidis@zf.com

Dr. rer. nat. Oliver Maria Kind, ZF, Automated Driving & Integral Cognitive Safety, 40547 Düsseldorf, Deutschland, oliver.kind@zf.com

Univ.-Prof. Dr.-Ing. Prof. h.c. Dr. h.c. Torsten Bertram, TU Dortmund, Institut für Roboterforschung, Lehrstuhl für Regelungssystemtechnik, 44221 Dortmund, Deutschland, torsten.bertram@tu-dortmund.de

Kurzfassung

Moderne Fahrzeuge verfügen über eine Vielzahl von unterschiedlichen Sensoren, mit denen sie sowohl die Umwelt wahrnehmen als auch Informationen über die Eigenbewegung des Fahrzeugs erhalten. Die Odometriesensoren im Fahrzeug sind dabei Teil des Antiblockiersystems (ABS) beziehungsweise des Elektronischen Stabilitätsprogramms (ESP) und messen unter anderem die Fahrzeuggeschwindigkeit sowie die Gierrate des Fahrzeugs. Hiermit lässt sich mit Hilfe einfacher Koppelnavigation die Fahrzeugeigenbewegung bestimmen [1]. Die genannten Sensoren sind häufig anfällig für Messrauschen und Drifts, da sie meist auf Basis von kostengünstigen Halbleitersensoren für den Automotive-Markt produziert werden, wodurch die Güte der Eigenbewegungsschätzung reduziert wird. Um die Qualität der Schätzung zu verbessern, können exterozeptive Fahrzeugsensoren hinzugezogen werden, mit dessen Hilfe, auf Basis des erfassten *statischen* Umfelds des Fahrzeugs, eine Korrektur der Eigenbewegungsprädiktion vorgenommen werden kann. Neben Radar- und Kamerasensoren sind hierbei vor allem Lidarsensoren von großem Interesse, da sie im Vergleich gegenüber den anderen Sensoren eine erhöhte Genauigkeit in radialer und angularer Richtung besitzen. Die Eigenbewegungskorrektur wird in diesem Beitrag mit Hilfe eines *Simultaneous Localization And Mapping* (SLAM)-Ansatzes durchgeführt, wobei detektierte statische Objekte eines Automotive-Lidarsensorsystems, bestehend aus sechs Ibeo LUX Sensoren mit 360°-Rundumsicht, als Landmarken verwendet und in einer lokalen temporären Karte des direkten Fahrzeugumfelds über einen limitierten Zeitraum geführt werden. Diese lokale Karte wird nur vom SLAM-Algorithmus selbst zur verbesserten Eigenbewegungsschätzung verwendet und es wird keine globale Karte erzeugt. Die Schätzung erfolgt dabei in 2D, das heißt die Pose des Fahrzeugs wird mittels der Position in x - und y -Richtung sowie der Orientierung relativ zu einem globalen festen Bezugssystem definiert. Als statische Objekte werden zum einen punktförmige Infrastrukturelemente (P_i) wie beispielsweise Straßenschilder oder Laternenpfosten am Straßenrand und linienförmige Objekte (L_i) wie Leitplanken oder Hauswände aus den detektierten Objekten des Lidarsensorsystems extrahiert. Dies ist noch einmal in Abbildung 1 verdeutlicht.

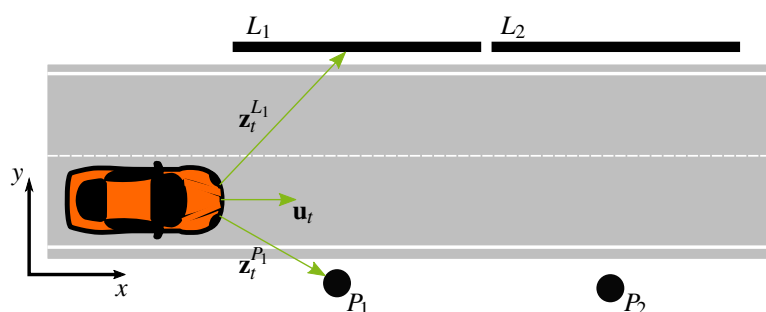


Abbildung 1 Darstellung der detektierten statischen Objekte für die Eigenbewegungskorrektur.

Das SLAM-Problem lässt sich mit Hilfe eines Dynamischen Bayesischen Netzwerks (DBN) veranschaulichen (Abb. 2). Hierbei stellt $\mathbf{x}_t$ die Fahrzeugpose zum aktuellen Zeitpunkt t dar, $\mathbf{u}_t$ definiert die Bewegungsinformation zwischen Zeitschritt $t - 1$ und t und $\mathbf{z}_t$ beschreibt die Messungen der Landmarken in der Umgebung $\mathbf{m}$ zum Zeitpunkt t . Die farbigen Knoten in der Abbildung 2 verdeutlichen, dass es sich dabei um vom Fahrzeug direkt gemessene Größen und bei den weißen Knoten um versteckte Variablen handelt, die es zu bestimmen gilt.

In der Literatur werden unterschiedliche Verfahren zum Lösen des SLAM-Problems vorgeschlagen [2]. Die Autoren von [3] präsentieren einen *Observability-Constrained Unscented Kalman-Filter* (OCUKF) mit einer verbesserten Sigmapunkt-Auswahlstrategie. Hierbei wird der Zustandsvektor $\mathbf{x}_t$ um die Positionen der Landmarken $\mathbf{m}_i$ erweitert. Im

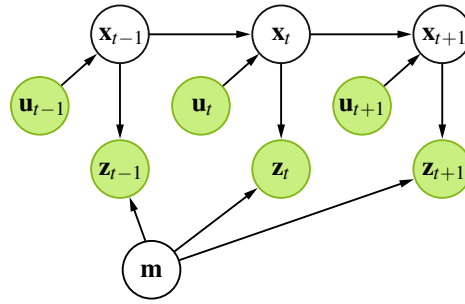


Abbildung 2 Dynamisches Bayesisches Netzwerk (DBN) des SLAM-Verfahrens nach [2].

Prädiktionsschritt wird zunächst basierend auf der Odometriemessung $\mathbf{u}_t$ und des Bewegungsmodells $f(\mathbf{x}_t, \mathbf{u}_t)$ der Zustand $\hat{\mathbf{x}}_{t+1}$ für den nächsten Zeitschritt geschätzt. Anschließend erfolgt auf Basis der Beobachtungen $\mathbf{z}_{t+1}$ der Umwelt $\mathbf{m}$ und des Messmodells $h(\hat{\mathbf{x}}_{t+1})$ der Korrekturschritt des Filters. Hierbei wird die Bestimmung der Messjacobimatrix so angepasst, dass die gewünschten Beobachtbarkeitseigenschaften des zugrunde liegenden nichtlinearen SLAM-Systems erhalten bleiben (*observability-constrained*), welches zu einer deutlich verbesserten Güte und Konsistenz des OCUKF im Vergleich zu einem Standard *Extended Kalman-Filter* (EKF)- oder UKF-basierten Verfahren führt.

Eine alternative Methode zum Lösen des SLAM-Problems basiert auf der Theorie der Faktorgraphen [4]. Dabei wird das DBN des SLAM-Problems (Abb.2) zum Vorbild genommen. Die Landmarken und Fahrzeugposen entsprechen den Knoten im Faktorgraph und somit die zu bestimmenden Variablen des zugrunde liegenden Gleichungssystems. Die Knoten sind über Kanten, welche auch als Faktoren bezeichnet werden, miteinander verknüpft und stellen weiche Nebenbedingungen zwischen den Variablen dar. Jeder Faktor entspricht einem (meist quadrierten) Fehlerterm basierend auf den Sensormessungen $(\mathbf{u}_t, \mathbf{z}_t)$ und dem zugehörigen Modell. Dabei muss nicht explizit zwischen einem Bewegungs- und einem Messmodell unterschieden werden, wie es beim Kalman-Filter der Fall ist. Die Bestimmung der Fahrzeugposen erfolgt anschließend über eine Optimierung des konstruierten nichtlinearen *Least Squares* (NLS)-Problems. In der Literatur werden verschiedene Verfahren zum effizienten Lösen des Graphen-basierten SLAM-Problems vorgestellt [5, 6, 7].

In diesem Beitrag liegt der Fokus auf dem Vergleich zwischen dem OCUKF und einem optimierungsbasierten SLAM-Verfahrens zur Verbesserung der Eigenbewegungsschätzung mittels detektierter statischer Objekte. Die Autoren von [3] zeigen bereits einen Vergleich beider Verfahren für einen öffentlich zugänglichen Datensatz, wobei der OCUKF eine leicht verbesserte Positionierungsgenauigkeit bei vergleichbarer Rechenzeit liefert. Im vorliegenden Beitrag werden beide Verfahren für einen selbst erstellten Datensatz, welcher sowohl urbane als auch Autobahn-Szenarien beinhaltet, miteinander verglichen, um eine Aussage über deren Eignung im realen Straßenverkehr zu treffen. Referenzpositionierungsdaten werden dabei mit Hilfe eines *Real Time Kinematic* (RTK)-GPS Systems mit Zentimetergenauigkeit generiert.

Literatur

- [1] Thrun, Sebastian, Wolfram Burgard, und Dieter Fox. *Probabilistic robotics*. MIT press, 2005.
- [2] Stachniss, Cyrill, John J. Leonard, und Sebastian Thrun. „Simultaneous localization and mapping.“ *Springer Handbook of Robotics*. Springer, Cham, 2016. 1153-1176
- [3] Huang, Guoquan P., Anastasios I. Mourikis, und Stergios I. Roumeliotis. „A quadratic-complexity observability-constrained unscented Kalman filter for SLAM.“ *IEEE Transactions on Robotics* 29.5 (2013): 1226-1243.
- [4] Kschischang, Frank R., Brendan J. Frey, und H-A. Loeliger. „Factor graphs and the sum-product algorithm.“ *IEEE Transactions on information theory* 47.2 (2001): 498-519.
- [5] Kümmerle, Rainer, et al. „g²o: A general framework for graph optimization.“ *Robotics and Automation (ICRA), 2011 IEEE International Conference on*. IEEE, 2011.
- [6] Kaess, Michael, et al. „iSAM2: Incremental smoothing and mapping using the Bayes tree.“ *The International Journal of Robotics Research* 31.2 (2012): 216-235.
- [7] Kaess, Michael, Ananth Ranganathan, und Frank Dellaert. „iSAM: Incremental smoothing and mapping.“ *IEEE Transactions on Robotics* 24.6 (2008): 1365-1378.

Feedforward Control of a Crane Manipulator with Four Load Chains

Michael Stoltmann, Christoph Woernle, University of Rostock, Chair of Technical Dynamics, 18059 Rostock

Deutschland, {michael.stoltmann,woernle}@uni-rostock.de

Pascal Froitzheim, Normen Fuchs, Wilko Flügge, Fraunhofer Research Institution for Large Structures in Production Engineering IGP, 18059 Rostock, Deutschland

Abstract

Metal sheets can be formed by means of a so-called ship building press in a cold plastic forming process. For positioning under the forming tool, the metal sheet is suspended by four chain hoists mounted on trolleys of an overhead crane with two bridges each with two trolleys. Bridges, trolleys and chain hoists are independently controllable to achieve the desired workpiece position. In addition to the flexibility of the workpiece itself the four chain attachment points at the plate are flexible in order to isolate the crane gear from process forces.

For controller design a simple model of the flexible plate with one elastic degree of freedom η , describing only the torsional deflection of the metal sheet, is used. Compared to other occurring deflection modes of the sheet, the torsional mode is predominant for the suspension chain force distribution. The controller design model is schematically shown in Fig. 1.

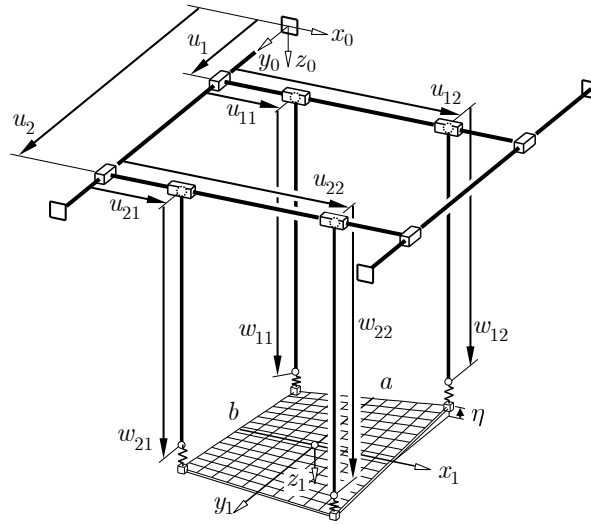


Fig. 1 Cable manipulator with four load chains suspending a flexible rectangular metal sheet.

The objective of the flatness-based control is to determine the actuator coordinates

$$\mathbf{u} = [u_1 \ u_2 \ u_{11} \ u_{12} \ u_{21} \ u_{22}]^T \quad \text{and} \quad \mathbf{w} = [w_{11} \ w_{12} \ w_{21} \ w_{22}]^T \quad (1)$$

of the crane in such a way that the metal sheet is moved between predefined rest positions without undesired sway motions. For this purpose, the dynamics of the system has to be taken into account. The position coordinates are summarised in the 7-vector

$$\hat{\mathbf{r}} = [\mathbf{r}_1^T \ \boldsymbol{\varphi}^T \ \eta]^T \quad (2)$$

as well as their time derivatives $\dot{\hat{\mathbf{r}}}$ and $\ddot{\hat{\mathbf{r}}}$. In [1] the nonlinear equations of motion of the crane manipulator were derived in the form

$$\mathbf{M}\dot{\hat{\mathbf{r}}} = \mathbf{k}^c(\dot{\hat{\mathbf{r}}}) + \mathbf{k}^e(\hat{\mathbf{r}}, \mathbf{u}, \mathbf{w}) + \mathbf{k}^{el}(\hat{\mathbf{r}}). \quad (3)$$

The mass matrix $\mathbf{M}$ and the generalised centrifugal and Coriolis forces $\mathbf{k}^c$ from (3) have the structure

$$\mathbf{M} = \begin{bmatrix} m\mathbf{E} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \boldsymbol{\Theta}_1 & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \frac{1}{9}m \end{bmatrix}, \quad \mathbf{k}^c = \begin{bmatrix} \mathbf{0} \\ -\dot{\boldsymbol{\omega}}\boldsymbol{\Theta}_1\boldsymbol{\omega} \\ 0 \end{bmatrix}. \quad (4)$$

The vector of generalised external forces on the plate $\mathbf{k}^e$ in (3) can be written as

$$\mathbf{k}^e = [\mathbf{k}_f^e \ \mathbf{k}_\tau^e \ \mathbf{k}_\eta^e]^T. \quad (5)$$

Here $\mathbf{k}_f^e$ is the resulting force of the gravity force and the four suspension forces, $\mathbf{k}_\tau^e$ is the resulting moment of the suspension forces and $\mathbf{k}_\eta^e$ is the generalised force on the elastic coordinate η due to the suspension forces. The generalised elastic force of the plate $\mathbf{k}^{el}$ is related to the bending stiffness of the plate K under consideration of its length a , width b as well as thickness h , Young's modulus E and Poisson's ratio ν [2],

$$\mathbf{k}^{el} = [\mathbf{0} \quad \mathbf{0} \quad k_\eta^{el}]^T \quad \text{with} \quad k_\eta^{el} = -c_f \eta, \quad c_f = \frac{32K(1-\nu)}{ab}, \quad K = \frac{Eh^3}{12(1-\nu^2)}. \quad (6)$$

The flatness property of the system is exploited to derive a feedforward control model. A flat system has the property that the state variables and control inputs can be algebraically expressed in terms of the so-called flat outputs and their time derivatives [3]. According to (1) and (2) there are ten independently controllable actuator coordinates but only seven position coordinates, the six rigid-body coordinates and the coordinate η describing the elastic degree of freedom of the plate. Thus, the problem is kinematically triple redundant. Technically advantageous solutions can be found by optimisation, for example by minimising the chain lengths.

A general movement in space and the results for the ten actuator coordinates are shown in Fig. 2. The metal sheet is moved along a desired trajectory within a given time interval, and the actuator coordinates are algebraically calculated from the equations of motion from (3).

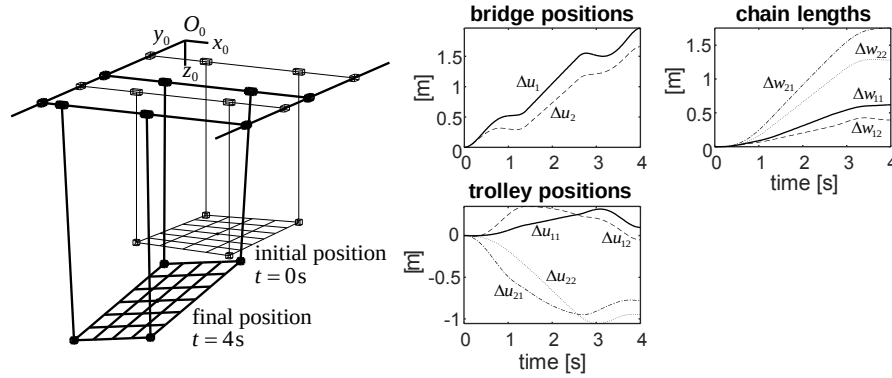


Fig. 2 Motion between rest positions and time trajectories of the actuator coordinates.

The realization of the feedforward controller on the basis of the nonlinear equations of motion requires the iterative solution of a nonlinear constrained optimization problem being computationally expensive and not directly suitable for implementation into a plant controller. Therefore the linearized equations of motion

$$\mathbf{M}\Delta\ddot{\mathbf{r}} + \mathbf{K}\Delta\mathbf{r} = \mathbf{H}_u\Delta\mathbf{u} + \mathbf{H}_w\Delta\mathbf{w} \quad (7)$$

with the actuator coordinates $\mathbf{u}$ (bridges and trolleys) and $\mathbf{w}$ (chain lengths) from (1) were formulated. They are obtained by successively linearizing all calculations during formulation of the nonlinear equations of motion (3).

The results for the actuator coordinates calculated from the nonlinear and also from the linearized equations of motion were transmitted to a Simpack multibody model of the crane manipulator. The system reaction on the determined actuator coordinates was satisfactorily simulated, resulting in only minor sway motions. A prototype system being under construction will be used for experimental validations.

Literatur

- [1] Stoltmann, M., Froitzheim P., Fuchs, N., Woernle, C.: *Flatness-Based Feedforward Control of a Crane Manipulator with Four Load Chains*. In: Corves, B. et al. (Eds.): EuCoMeS 2018, pp. 61-68, Springer, 2019.
- [2] Bhaskar, K., Varadan, T.: *Plates*. John Wiley Sons (2014).
- [3] Fliess, M., Lévine, J., Martin, P., Rouchon, P.: *Flatness and defect of nonlinear systems: introductory theory and examples*. Int. J. Control **51**, 1327-1361 (1995).

Weiterentwicklung eines Model-Update-Ansatzes flexibler Mehrkörpersysteme

Andreas Schulze, M.Sc., Univ.-Prof. Dr. Ing. habil. Christoph Woernle, Universität Rostock, Lehrstuhl für Technische Mechanik/Dynamik, 18059 Rostock, Deutschland, {andreas.schulze2, woernle}@uni-rostock.de
Dr. Ing. habil. János Zierath, W2E Wind to Energy GmbH, 18055, Rostock, Deutschland, jzierath@wind-to-energy.de

Kurzfassung

In den letzten zwei Jahrzehnten hat sich die Methode der flexiblen Mehrkörpersysteme als Standardverfahren für die effiziente und zuverlässige numerische Simulation komplexer mechanischer Systeme etabliert. Insbesondere für Untersuchungen, in denen Komponenten des mechanischen Systems großen, nichtlinearen Bewegungen ausgesetzt sind, während die Verformung der Komponenten klein bleibt, ist die flexible Mehrkörpersimulation (FMKS) anderen gängigen Ansätzen, wie z. B. der Finite-Elemente-Methode (FEM), überlegen.

In FMKS werden flexible und starre Körper über kinematische Zwangsbedingungen oder über klassische Kraftgesetze miteinander verbunden, um das dynamische Verhalten der Systeme unter äußeren Lasten zu untersuchen. In vielen Fällen, wie beispielsweise bei der Untersuchung großer und schlanker mechanischer Strukturen, ist die Implementierung der flexiblen Komponenten entscheidend für die Gesamtqualität der Simulation. Der allgemeine Ansatz zur Einbettung flexibler Körper in eine Mehrkörperumgebung umfasst drei aufeinanderfolgende Schritte [1, 2]: Diskretisierung der elastischen Struktur mittels FEM, Reduktion der Freiheitsgrade (FHG) mittels Model-Order-Reduction (MOR)-Techniken, Kopplung der elastischen Unterstruktur an die Mehrkörperumgebung mittels Floating-Frame-of-Reference (FFR). Obwohl es sich um die Standardmethode gängiger Mehrkörpersoftware handelt, besitzt diese Vorgehensweise mehrere Nachteile, insbesondere wenn die Steifigkeits, Massen und Dämpfungseigenschaften der flexiblen Komponente entweder nicht genau bekannt sind oder nicht ohne erheblichen Rechenaufwand modelliert werden können.

Die Problematik von Modellunsicherheiten betrifft prinzipbedingt alle numerischen Modellierungsansätze. Vor diesem Hintergrund wurden in den letzten Jahrzehnten erhebliche Anstrengungen unternommen, die Qualität der numerischen Simulation zu verbessern, indem experimentelle Daten eines realen Prototyps bei der Modellbildung verwendet werden. Die zahlreichen Methoden, die in diesem Zusammenhang entwickelt wurden, werden als Modell Update (MU) bezeichnet. Eine umfassende Übersicht über dieses Forschungsgebiet ist in [3] enthalten. Die Grundidee des MU besteht in der gezielten Anpassung eines vordefinierten Sets von Modellparameter oder direkt der Systemmatrizen auf Basis eines Vergleichs zwischen dem numerischen Modell und experimentellen Ergebnissen.

Ein besonderes Interesse in der Entwicklung von MU-Verfahren lag auf der Verbesserung der Korrelation zwischen einer realen Struktur und einem entsprechenden FEM-Modell, siehe z. B. [4, 5]. Die direkte Anwendung dieser Verfahren auf FMKS-Modelle ist jedoch mit erheblichen Einschränkungen verbunden. Der wesentliche Nachteil des MU in Bezug auf die FMKS liegt in der Verwendung modaler Parameter (Eigenformen, Eigenfrequenzen und modale Dämpfung) und dem damit verbundenen Eigenwertproblem. Dies führt zwangsläufig zu der Vernachlässigung des zumeist stark nichtlinearen transienten Verhaltens der FMKS-Systeme. Aus diesem Grund erfolgten nur sehr wenige Forschungsaktivitäten in der direkten Anwendung von MU auf die FMKS, siehe [6]. Neben der direkten Anwendung der MU-Verfahren auf ein FMKS-Modell besteht die Möglichkeit eines indirekten MU. Hierbei werden die flexiblen Komponenten einzeln und noch vor der Einbindung in die FMKS, d. h. auf Ebene der FEM, einem Model Update unterzogen. Dieses Vorgehen entspricht der konventionellen MU-Methode, wohingegen das nichtlineare Verhalten der Gesamtstruktur, unter der Voraussetzung kleiner, linearer Verformungen der einzelnen Komponenten, nach dem Zusammenbau erhalten bleibt. Die entscheidenden Nachteile sind der erhebliche numerische Aufwand verbunden mit dem MU der jeweiligen flexiblen Systemkomponente, die damit verbundenen numerischen Fehler des MU und die numerischen Fehler infolge des MOR Prozesses.

Die nachteiligen Auswirkungen konventioneller MU-Verfahren in Anwendung auf die FMKS rechtfertigen die Notwendigkeit eines alternativen Ansatzes zur Verbesserung der Korrelation zwischen einem mechanischen System und einem entsprechenden FMKS-Modell. Einen wesentlichen Schritt in diese Richtung liefert die Forschungsarbeit in [7]. Das dort vorgestellte Verfahren ermöglicht die direkte Modellierung der flexiblen Systemkomponenten auf Basis experimentell ermittelter Eigenformen, Eigenfrequenzen und Dämpfungseigenschaften. Ein MU des FMKS-Modells erfolgt somit über die direkte Einbindung der experimentellen Daten in die systembeschreibenden Bewegungsdifferentialgleichungen. Auf ein vorheriges MU der Komponenten auf FE-Ebene kann damit verzichtet werden.

Der vorliegende Beitrag untersucht das in [7] vorgestellte Verfahren am Beispiel eines einfachen elastischen Balkens in frei-frei Konfiguration und als Kragbalken. Der prinzipielle Verfahrensablauf des MU-Ansatzes ist in Abb. 1 gegeben. Den Ausgangspunkt der Untersuchungen bildet der Messaufbau. Beide Messaufbauten werden mit Beschleunigungsaufnehmern ausgestattet und über einen Impulshammer zu Schwingungen angeregt. Anschließend werden die gemessenen Daten analysiert und die modalen Eigenschaften der Struktur unter Verwendung von Verfahren der experimentellen Mo-

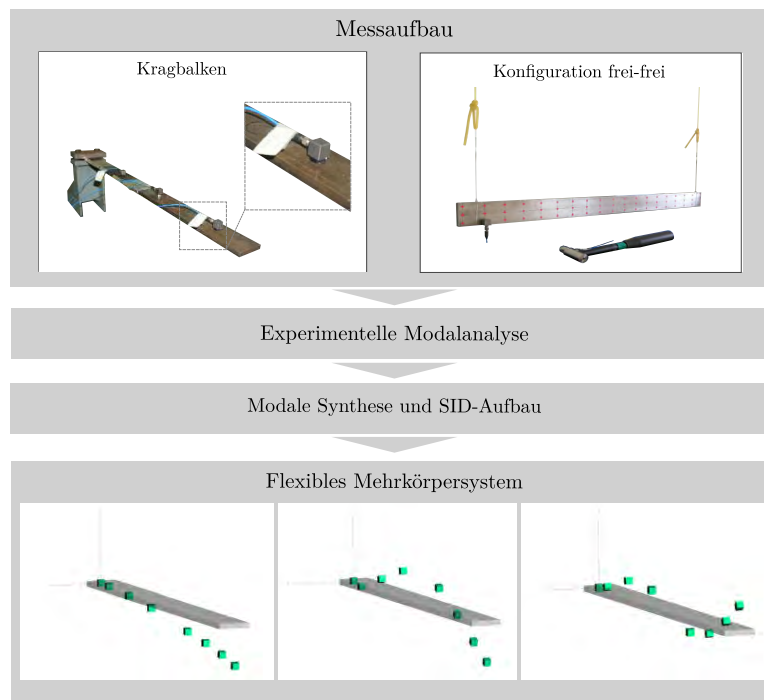


Bild 1 Prinzipieller Verfahrensablauf zum Model-Update flexibler Mehrkörpersysteme

Modalanalyse identifiziert. Im nächsten Schritt erfolgt die direkte Synthese des flexiblen Balkenmodells unter Verwendung der gemessenen modalen Größen gemäß [7]. Die Einbindung des flexiblen Körpers in die Mehrkörpersoftware SIMPACK erfolgt unter Verwendung des Standard-Input-Data-Formats (SID) [8]. Erste Ergebnisse bezüglich des Kragbalkens zeigen in Übereinstimmung mit den Ergebnissen aus [7] ein identisches Deformationsverhalten des realen Kragbalkens und des zugehörigen FMKS-Modells. Weiterführende Untersuchungen der systembeschreibenden Bewegungsdifferentialgleichungen weisen jedoch Abweichungen in den Starrkörpereigenschaften (Masse und Massenträgheitsmomente) und den Kopplungstermen zwischen elastischer Deformation und Starrkörperbewegung auf. Vor diesem Hintergrund wird eine Erweiterung des Verfahrens aus [7] angestrebt, indem die Abweichungen in den Starrkörpereigenschaften unter Verwendung der FE-Massenmatrix des Balkens minimiert werden.

Literatur

- [1] Shabana, A. A.: *Dynamics of multibody systems*. Cambridge: Cambridge University Press, 2014.
- [2] Schwertassek, R.; Wallrapp, O.: *Dynamik flexibler Mehrkörpersysteme*. Wiesbaden: Vieweg+Teubner Verlag, 1999.
- [3] Mottershead, J. E.; Friswell, M. I.: *Model Updating In Structural Dynamics: A Survey*. Journal of Sound and Vibration.(1993) 167-2, S. 347-375.
- [4] Živanović, S.; Pavic, A.; Reynolds, P.: *Finite element modelling and updating of a lively footbridge: The complete process*. Journal of Sound and Vibration.(2007) 301-2, S. 126-145.
- [5] Minas, C.; Inman, D. J.: *Matching Finite Element Models to Modal Data*. Journal of Vibration and Acoustics.(1990) 112-1, S. 84.
- [6] Manzato, S.; Peeters, B.; Toso, A.; van der Auweraer, H.; Osgood, R.: *Model updating methodologies for multibody simulation models: application to a full-scale wind turbine*. New York: Springer (Conference Proceedings of the society for Experimental Mechanics Series), 2011.
- [7] Lein, C.; Woller, J.; Hopf, H.; Beitelschmidt, M.: *Approach for Modelling Flexible Bodies based on Experimental Data with Utilization in Elastic Multibody Simulation*. ECCOMAS Thematic Conference on Multibody Dynamics, Prag, 2017.
- [8] Wallrapp, O.: *Standardization of Flexible Body Modeling in Multibody System Codes, Part I: Definition of Standard Input Data*. Mechanics of Structures and Machines. (1994) 22-3, S. 283-304.

Parameteridentifikation eines dynamischen mechanischen Systems mittels Frequenzganganalyse und rekursiver Partikelschwarmoptimierung für die Trajektorienplanung

Christian Gollee, TU Dresden, Institut für Naturstofftechnik, 01062 Dresden, Deutschland, christian.gollee@tu-dresden.de

Kurzfassung

Mechanismen sind in der industriellen Praxis allgegenwärtig. Sie dienen dazu, die Übertragung von Kräften und Bewegung auch bei hohen Betriebsgeschwindigkeiten präzise zu realisieren. Parallele Mechanismen eignen sich aufgrund ihrer hohen Steifigkeit und des geringen Anteils an bewegten Massen besonders für die Aufgabe des Produkthandlings. Dennoch treten aufgrund der immer vorhandenen Elastizität sowohl der Mechanismenglieder als auch der Maschinengestelle zwangsläufig Schwingungen auf. Neben unangenehmen Erscheinungen wie Lärm und Verschleiß sind die damit verbundenen Bewegungsabweichungen, z.B. bei der Handhabung von Produkten, besonders störend. Die möglichen Schäden reichen von einer ungenauen Positionierung des Produkts bis hin zum Verlust während der Bewegung.

Die genaue Kenntnis der dynamischen Eigenschaften des Systems ermöglicht ein Bewegungsdesign, das zu einer geringeren Schwingungsanregung führt. Mit geeigneten Algorithmen und einem mathematischen Modell (Bild 1) des technischen Systems ist es möglich, schwingungsoptimale Bewegungen unter Einhaltung vorgegebener Einschränkungen automatisch zu erzeugen. Voraussetzung ist, dass alle unbekannten Modellparameter ermittelt werden. Parameteridentifikationsmethoden haben das Ziel, die Systemmatrizen des Modells aus gemessenen dynamischen Daten zu identifizieren [1]. Es gibt eine ganze Reihe von Methoden, um eine mathematische Beschreibung für ein bestimmtes System zu erhalten [2]. In den meisten Fällen wird mit einem Hammer oder Shaker eine definierte Erregerkraft aufgebracht und die Antwort des untersuchten Systems mit Sensoren gemessen. Um zu einer vollständigen Charakterisierung zu gelangen, muss entweder der Punkt der Anregung oder der Ort der gemessenen Antwort über das System [3] laufen. Dieser Beitrag stellt den Versuch dar, ein dynamisches mechanisches System mit Hilfe einer Methode zu charakterisieren, die minimale Ressourcen, in Bezug auf Messtechnik und Rechenaufwand, erfordert. Zu diesem Zweck wird für das untersuchte System ein Modell erstellt, das Verfahren zur Berechnung der Starrkörperdynamik mit Methoden zur kinematischen Analyse von Mechanismen kombiniert. Es wird ein Bewegungsplanungswerkzeug entworfen, das die Integration des Motion-Control-Systems in die Entwicklungsumgebung des Algorithmus über eine spezielle Schnittstelle ermöglicht. Dadurch können ein adaptives und effizientes Bewegungsdesign durchgeführt und Messdaten unkompliziert und schnell erfasst, dargestellt und verarbeitet werden (Bild 2).

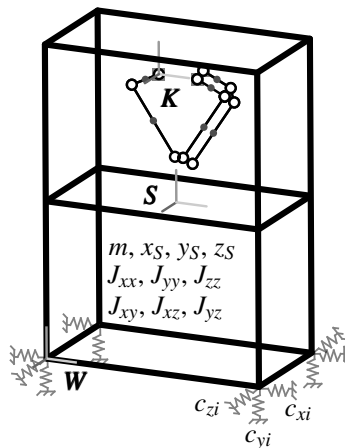


Bild 1 Abstraktes Modell des Gesamtsystems

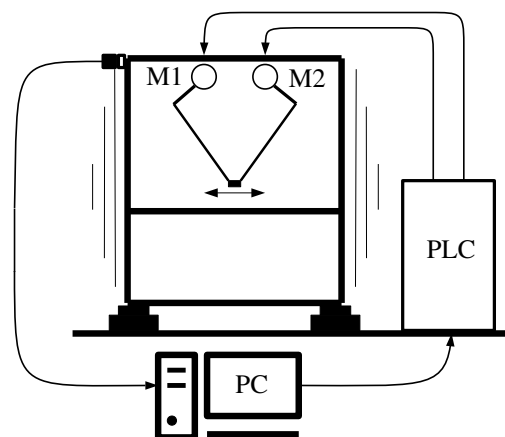


Bild 2 Versuchsstand

Literatur

- [1] Link, M.: *Theory of a method for identifying incomplete system matrices from vibration test data* Zeitschrift für Flugwissenschaften und Weltraumforschung 9. (1985)
- [2] Ewins, D.J.: *Modal testing: theory, practice, and application* Research Studies Press, 2000.
- [3] Avitabile, P.: *Experimental modal analysis - A simple non-mathematical presentation* Sound and Vibration Magazine 35. (2001), S. 20-31.

Analyse der Benutzer- und Softwareanforderungen bei der Entwicklung eines modularen Software-Frameworks für mechanische Systeme am Beispiel von Fahrdynamikmodellen

Analysis of User and Software Requirements in the Development of a Modular Software Framework for Mechanical Systems based on Vehicle Dynamics Models

Janosch Luttmer M.Sc., Universität Duisburg-Essen, Lehrstuhl für Mechatronik, 47057 Duisburg, Deutschland, janosch.luttmer@stud.uni-due.de;

Sebastian Reicherts M.Sc., Universität Duisburg-Essen, Lehrstuhl für Mechatronik, 47057 Duisburg, Deutschland, sebastian.reicherts@uni-due.de;

Prof. Dr.-Ing. Dr. h.c. Dieter Schramm, Universität Duisburg-Essen, Lehrstuhl für Mechatronik, 47057 Duisburg, Deutschland, dieter.schramm@uni-due.de.

Kurzfassung

Im vorliegenden Beitrag wird eine universell anwendbare, strukturierte Entwicklungsumgebung zum Erstellen und Modifizieren von Fahrdynamikmodellen präsentiert. Sie ermöglicht ein komponentenbasiertes Erstellen von Fahrdynamikmodellen unterschiedlicher Komplexität und bietet ferner eine geeignete Verwaltung der Komponenten. Der vorliegende Beitrag zielt weniger auf die Präsentation der Funktionalitäten des erstellten Frameworks ab, welches speziell auf die Anforderungen und Wünsche des Lehrstuhls für Mechatronik der Universität Duisburg-Essen zugeschnitten ist. Hier soll vielmehr auf den Entwicklungsprozess eingegangen werden, der auf vergleichbare Fragestellungen anwendbar ist. Dieser orientiert sich an einem um iterative Ansätze erweiterten V-Modell, wobei die Analyse der Benutzer- und Softwareanforderungen von besonderem Interesse ist. Die Anforderungsanalyse wird anhand des Volere-Prozesses durchgeführt, welcher die Erstellung einer umfangreichen Anforderungsspezifikation unterstützt. Diese dient zum einen als Entwicklungsleitfaden und kann zum anderen um Änderungswünsche und neu auftretende Anforderungen ergänzt werden.

Abstract

This paper presents a universally applicable, structured development environment for creating and modifying vehicle dynamics models. It enables the component-based creation of vehicle dynamics models with varying complexity and offers appropriate management of the components. The goal of this contribution is not the detailed presentation of the functionalities of the created framework, which is specially tailored to the requirements and wishes of the Chair of Mechatronics of the University of Duisburg-Essen. Instead, the development process, which is applicable to comparable questions, should be considered here. It is based on the V-model, which is supplemented by iterative approaches. In this context, the analysis of user and software requirements is of particular interest. It is performed using the Volere requirements process, which supports the creation of an extensive requirement specification. The specification can be used as a development guideline and allows the integration of change requests and new requirements.

1 Einleitung

Simulationstechniken nehmen eine zunehmend größere Rolle im Entwicklungsprozess vieler Industriezweige ein. Die Automobilindustrie bildet hier keine Ausnahme. Besonders in den frühen Entwicklungsphasen erlauben Simulationen belastbare Aussagen über die Funktionsweise von Bauteilen. Eine Möglichkeit ist hierbei der Einsatz von Fahrsimulatoren, die den realen Fahrer in eine interaktive Simulationsumgebung einbinden und insbesondere der Erforschung der Mensch-Maschine-Schnittstelle dienen [1]. Während zur Entwicklung detaillierter Fahrdynamikmodelle und einzelner Komponenten zahlreiche kommerzielle Werkzeuge existieren, bedarf es vor dem Hintergrund ei-

nes wachsenden Anteils an modellbasierten Simulationstechniken wie MATLAB/Simulink in der automobilen Entwicklung vor allem Unterstützung bei der Verwaltung sowie dem Austausch von Modellen und deren Komponenten. Das Interesse an einem Werkzeug zur Strukturierung des Entwicklungsprozesses besteht jedoch nicht nur in der Automobilindustrie, sondern in zahlreichen teambasiert arbeitenden Entwicklungsabteilungen vieler Industriezweige. Hierzu zählt auch der Lehrstuhl für Mechatronik der Universität Duisburg-Essen. Am Lehrstuhl werden seit vielen Jahren Fahrdynamikmodelle unterschiedlicher Komplexität entwickelt und an Fahrsimulatoren erprobt. Die Modelle und deren Komponenten werden sowohl von Mitarbeitern als auch von Studenten erstellt und weiter-

entwickelt, was den im Entwicklungsprozess involvierten Personenkreis in die Höhe treibt und unweigerlich zu einer starken Fluktuation im Selbigen führt. Da hier jedoch keine einheitlichen Richtlinien existieren, sind die Entwicklungstätigkeiten mit einem hohen Aufwand verbunden und führen zu unübersichtlichen und schwer nachvollziehbaren Modellen.

Das hier vorgestellte Framework dient der Entwicklung von Fahrdynamikmodellen für den Einsatz in Fahrsimulatoren. Es soll am Lehrstuhl für Mechatronik der Universität Duisburg-Essen eingesetzt und von Mitarbeitern und Studenten im gleichen Maße genutzt werden. Hierzu wird dem Anwender nicht nur eine anpassbare Architektur, sondern zusätzlich eine Sammlung von Richtlinien und Lösungen zur strukturierten Erstellung von Fahrdynamikmodellen bereitgestellt.

Zu diesem Zweck wird im vorliegenden Beitrag gezeigt, wie der Entwicklungsprozess eines Frameworks anhand eines geeigneten Vorgehensmodells strukturiert sein kann. Der Fokus liegt hierbei auf der Analyse der Benutzer- und Softwareanforderungen, die zu Projektbeginn durchgeführt und durch geeignete Prozesse und Werkzeuge unterstützt wird. Darüber hinaus werden Maßnahmen zur Integration von Änderungswünschen sowie zur Anforderungsverfolgung dargelegt. Den Abschluss bildet eine Zusammenfassung des Beitrags sowie ein Ausblick auf weiterführende Forschungstätigkeiten.

2 Stand der Technik

Frameworks bieten ein mächtiges Werkzeug, die Entwicklungstätigkeiten zu strukturieren und unternehmensübergreifende Standards zu setzen. Im Jahr 2014 wurde das von der Argonne National Laboratory in Kooperation mit General Motors entwickelte Werkzeug *Autonomie* vorgestellt [2]. Autonomie stellt ein MATLAB-basiertes Framework für die Entwicklung, Simulation und Analyse von Steuerungssystemen im Fahrzeug dar. Sein Ziel ist unter anderem das einfache und schnelle Erstellen von Modellen unterschiedlichen Komplexitäts- und Abstraktionslevels auf Basis einer Plug-and-Play-Architektur. Ein wichtiger Bestandteil ist hierbei die Parallelisierung der Entwicklungsarbeiten mehrerer Abteilungen. In [3] wird angemerkt, dass Automobilhersteller zahlreiche Abteilungen besitzen, die sich mit der Fahrdynamiksimulation auf unterschiedlichen Fahrzeugebenen befassen. Hierbei entwickelt und simuliert jede Abteilung eine Reihe eigener Fahrdynamikmodelle, wodurch ein einfacher und schneller Austausch zwischen den Abteilungen nicht gewährleistet ist. Dies führt letztendlich nicht nur zu einem Zeit-, sondern auch zu einem Qualitätsverlust. [3]

Im Hinblick auf den Anwendungszweck der vorliegenden Arbeit sei darauf hingewiesen, dass Autonomie nicht auf die Erstellung umfangreicher Simulinkmodelle zur Anwendung in Fahrsimulatoren ausgerichtet ist. Aus diesem Grund ist die Entwicklung eines geeigneten Frameworks zweckmäßig, dessen methodische Grundlagen Teil des folgenden Abschnitts sind.

3 Entwicklungsmethodik

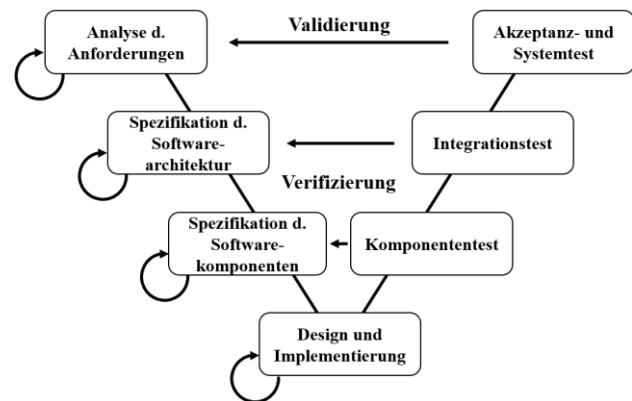


Bild 1 Entwicklung des Frameworks basierend auf einem modifizierten V-Modell, in Anlehnung an [4]

Im Sinne eines durchgängigen Softwareentwicklungsprozesses kommen Vorgehensmodelle zum Einsatz. Sie bieten eine zeitliche und inhaltliche Strukturierung und somit einen organisatorischen Rahmen für die Durchführung des Projekts. Aufgrund der Vielzahl unterschiedlicher Vorgehensmodelle kann eine Unterteilung in sequenzielle, iterative und agile Methoden vorgenommen werden. Während die erste Gruppe durch ein einmaliges Bearbeiten der anfallenden Aufgaben charakterisiert ist, beruhen iterative Modelle auf einer schrittweisen, in mehreren Zyklen durchzuführenden Entwicklung und ermöglichen eine verbesserte Reaktion auf Änderungen während des Entwicklungsprozesses. Um den Forderungen nach immer kürzeren Entwicklungszeiten bei häufig wechselnden Anforderungen gerecht zu werden, wurden letztlich agile Vorgehensmodelle wie Scrum oder Extreme Programming entwickelt. Sie zeichnen sich durch eine hohe Flexibilität und Schnelligkeit bei der Berücksichtigung von Änderungen aus und reduzieren den vergleichsweise hohen Dokumentationsaufwand anderer Vorgehensmodelle. [5]

Ein in der Praxis weit verbreitetes Vorgehensmodell ist das V-Modell, welches im Jahr 1979 von Barry Boehm explizit hinsichtlich der Softwareentwicklung erstellt wurde und sich den sequenziellen Methoden zuordnen lässt. Es ist durch die Anordnung von konstruktiven Tätigkeiten auf der linken sowie prüfenden Vorgängen auf der rechten Seite charakterisiert und misst demnach dem Entwurf und dem Test eine gleichwertige Bedeutung bei. Jedoch impliziert dies auch, dass in den Testphasen lediglich jene Fehler identifiziert werden, die in der symmetrischen Entwurfsphase, das heißt auf derselben Abstraktionsstufe entstanden sind. Daher sind Verfahren zur Fehlervermeidung und -behebung im frühen Entwicklungsstadium von besonderer Bedeutung. [5]

Aufgrund der hohen praktischen Relevanz des V-Modells orientiert sich das hier beschriebene Entwicklungsvorhaben an einem in [4] vorgestellten und im Rahmen der automobilen Softwareentwicklung eingesetzten V-Modell, welches um iterative Ansätze ergänzt wird (vgl. **Bild 1**). Weitere Treiber hinsichtlich dieser Entscheidung sind der begrenzte Projektumfang sowie die geringe Anzahl der beteiligten Entwickler. Beginnend mit der Analyse der

Benutzer- und Softwareanforderungen wird zuerst die Softwarearchitektur spezifiziert. Dem schließt sich die detaillierte Betrachtung der Softwarekomponenten an. Nach der vollständigen Spezifikation folgen die Implementierung sowie die abschließenden Testphasen. Diese unterteilen sich in Komponenten-, Integrations- und Akzeptanzbeziehungswise Systemtest. Da wie bereits erwähnt die späte Berücksichtigung von Fehlern aus frühen Projektphasen zu den Limitationen des klassischen V-Modells gehören, werden zur Entwicklung des hier beschriebenen Frameworks Reviews während der konstruktiven Phasen integriert. Sie führen zu einem iterativen Durchlaufen eines Schritts, damit eine möglichst große Anzahl an Aspekten berücksichtigt und somit die Qualität des Phasenoutputs erhöht wird (vgl. **Bild 1**).

4 Analyse der Benutzer- und Softwareanforderungen

Um die Anforderungen an das zu entwerfende System frühzeitig zu beschreiben und während des Projektverlaufs zu verfolgen, kommt dem Requirements Engineering eine besondere Bedeutung zu. Es setzt sich aus den vier Haupttätigkeiten *Ermitteln*, *Dokumentieren*, *Prüfen* bzw. *Abstimmen* und *Verwalten* zusammen, wobei die drei erstgenannten unter der Anforderungsanalyse zusammengefasst werden können [6]. Zur Strukturierung der Aktivitäten während der Anforderungsanalyse wird hier der von [7] entwickelte Volere-Prozess eingesetzt, dessen methodische Grundlagen sowie dessen Umsetzung im Rahmen der Entwicklung des Frameworks Gegenstand des folgenden Abschnitts sind. Das Verwalten als weitere Kernaktivität wird auch als Requirements Management bezeichnet und befasst sich unter anderem mit der Anforderungsverfolgung und dem Änderungsmanagement, auf die sich Abschnitt 5 fokussiert [6].

4.1 Volere-Prozess

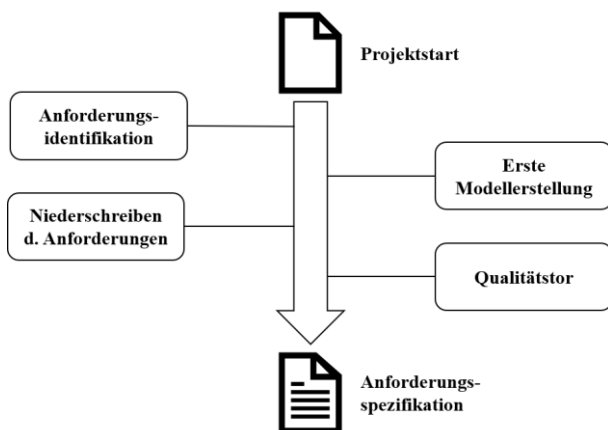


Bild 2 Volere-Prozess, in Anlehnung an [7]

Der Volere-Prozess setzt sich im Wesentlichen aus

- dem Identifizieren der Anforderungen,
- dem Niederschreiben der Anforderungen und
- der Qualitätskontrolle

zusammen (vgl. **Bild 2**). Hierbei lässt sich eine Unterteilung in funktionale und nicht-funktionale Anforderungen sowie Projektbeschränkungen vornehmen. Während funktionale Anforderungen fachlicher Natur sind, zu berücksichtigende Funktionen beschreiben und möglichst unabhängig von technologischen Festlegungen sein sollen, beziehen sich nicht-funktionale Anforderungen auf qualitative Eigenschaften wie Optik, Haptik, Nutzbarkeit, Performance oder Sicherheit. Letztlich unterliegt die Funktionalität des Endprodukts den Projektbeschränkungen, die zu einer Einengung des Lösungsraums führen. [7]

Die Aufgabe der Anforderungsidentifikation ist es, die Ziele und Anforderungen an das Projekt mit geringem Aufwand und angepasst an die Rahmenbedingungen zu erfassen. Hierbei steht vor allem der Kundennutzen im Vordergrund, weshalb der ausführlichen Kommunikation mit den Interessenvertretern (Stakeholdern) eine besondere Bedeutung zukommt. Zur Ermittlung der Anforderungen können verschiedenste Techniken eingesetzt werden. Hierzu zählen unter anderem Kreativitätstechniken, Befragungstechniken, das Erstellen von Use-Case-Modellen sowie das Beobachten bestehender Systeme. [6]

Die Sammlung potentieller Anforderungen wird im nächsten Schritt formalisiert und niedergeschrieben. Hierbei ist auf eine klare und konsistente Ausarbeitung zu achten. Dies äußert sich in einer formalisierten Beschreibung, so dass die Anforderungen frei von Widersprüchen und Redundanzen sind. Zudem sollen sie sich klar von der Lösungsbeschreibung abgrenzen und verifizierbar sein. Neben diesen Eigenschaften ist auf eine übersichtliche Organisation der Anforderungen zu achten, wobei geeignete Vorlagen verwendet werden können. [8]

Die im vorherigen Schritt formalisierten, potentiellen Anforderungen werden zuletzt einer Qualitätskontrolle unterzogen. Hier wird jede Anforderung individuell bezüglich der bereits beschriebenen und weiteren Attribute wie Messbarkeit oder Vollständigkeit getestet, um geeignete Anforderungen in die Anforderungsspezifikation zu überführen und abgelehnte Anforderungen an den Ersteller zur Klärung, Überarbeitung oder Eliminierung zurückzuleiten. Letzteres spiegelt auch den iterativen Charakter des Volere-Prozesses wider. Nach Beenden der Qualitätskontrolle sind die bewilligten Anforderungen in der Anforderungsspezifikation vereint. Sie stellt das finale Dokument des Volere-Prozesses dar und dient als Kommunikationsvehikel mit den Stakeholdern sowie als Leitfaden im weiteren Entwicklungsprozess. Darüber hinaus können Änderungswünsche eingepflegt werden, um der Dynamik des Entwicklungsprozesses gerecht zu werden. [7]

4.2 Identifikation der Anforderungen

Die Anforderungsidentifikation erfolgt mit Hilfe geeigneter Ermittlungstechniken, wobei unter Berücksichtigung der Projekteigenschaften und einer in [6] dargestellten Bewertungstabelle drei Techniken als zweckmäßig erscheinen. Hierzu zählen das Beobachten bestehender Systeme, Interviews mit den Stakeholdern sowie das Erstellen von Use-Case-Diagrammen.

Das Beobachten bestehender Systeme hat zum Ziel, sowohl bereits zufriedenstellende Lösungen als auch Optimierungspotenziale aufzufinden, welche im Rahmen der

Entwicklungsaufgabe berücksichtigt werden sollen [6]. Bezogen auf das in diesem Beitrag vorgestellte Entwicklungsvorhaben werden vor allem bestehende Fahrdynamikmodelle des Lehrstuhls für Mechatronik in *MATLAB/Simulink* untersucht, wobei Verbesserungen bezüglich der Übersichtlichkeit und Einheitlichkeit der Modelle zu erzielen sind. Dies betrifft unter anderem die Signalverwaltung, die Namensgebung von Modellblöcken sowie deren Farbgebung. Ein zweiter Untersuchungsgegenstand ist das in Abschnitt 2 beschriebene Entwicklungsframework *Autonomie*, welches die modulare und komponentenbasierte Modellierung von Fahrzeugen erlaubt und somit viele übertragbare Anhaltspunkte liefert.

Die Durchführung von Interviews mit den Stakeholdern bietet den Vorteil, die konkreten Wünsche einzelner Personen zu ermitteln und einen Überblick über das Entwicklungsvorhaben sowie dessen Projektziele zu bekommen [6]. Den Personenkreis der Stakeholder umfasst die Mitarbeiter des Lehrstuhls für Mechatronik sowie Studenten, die im Rahmen von Abschlussarbeiten oder Hilfstätigkeiten Fahrdynamikmodelle mit Hilfe des Frameworks entwickeln sollen. Hierbei ist die erstgenannte Gruppe von besonderem Interesse für die Durchführung der Interviews, da hier eine regelmäßige Verwendung des Frameworks vorliegt. Bezüglich der Studenten ist jedoch zu beachten, dass sie mitunter nicht über ein tiefgreifendes Fachwissen hinsichtlich komplexer Fahrdynamikmodelle verfügen und somit besondere Anforderungen an die Entwicklungsaufgabe verursachen. Hierzu zählen die Entwicklung eines umfassenden Regelwerks zur strukturierten Erstellung von Fahrdynamikmodellen sowie eine detaillierte Softwaredokumentation. Die Auswertung der Interviews zeigt, dass von Seiten der Stakeholder das Bedürfnis nach einem Framework besteht, welches

- eine universell anwendbare und strukturierte Entwicklungsumgebung zur Erstellung und Modifizierung von Fahrdynamikmodellen bereitstellt,
- ein komponentenbasiertes Entwickeln von Fahrdynamikmodellen unterstützt, um zu einer Steigerung der Flexibilität beizutragen,
- die Komponenten in einer geeigneten Weise verwaltet und
- den teambasierten Entwicklungsprozess fördert.

Die Wünsche, die im weiteren Projektverlauf als Entwicklungsziele betrachtet werden, erlauben im Anschluss die Ableitung konkreter Anforderungen, die sich vor allem auf die Projektbeschränkungen und nicht-funktionalen Anforderungen beziehen. Beispielhaft sei hier die Verwendung der Entwicklungsumgebung *MATLAB/Simulink* als Projektbeschränkung und die Übersichtlichkeit der im Framework erstellten Fahrzeugmodelle als nicht-funktionale Anforderung genannt.

Darüber hinaus dienen die Interviews dazu, die Hauptfunktionen des Frameworks (Use Cases) mit den Stakeholdern abzustimmen. Aus ihnen werden im weiteren Verlauf Use-Case-Diagramme erstellt, die die Identifikation von Anforderungen erleichtern und als Referenz beim späteren Niederschreiben dienen. Während sich die bereits beschriebenen Ermittlungstechniken vornehmlich auf Projektbeschränkungen und nicht-funktionale Anforderungen bezie-

hen, legt die Analyse der Use-Case-Diagramme den Fokus auf funktionale Anforderungen.

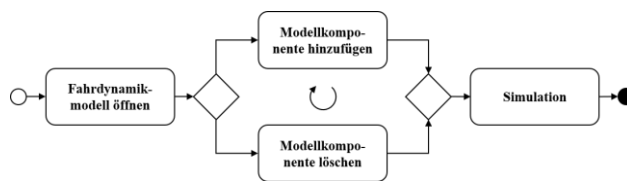


Bild 3 Use-Case-Diagramm *Modifizieren von bestehenden Fahrdynamikmodellen* (Use Case 1)

Bezüglich des Frameworks können acht Use Cases festgelegt werden. Sie umfassen das Erstellen, Modifizieren und Löschen von Fahrdynamikmodellen sowie deren Modellkomponenten. Ferner besteht eine Hauptfunktionalität darin, die Modellkomponenten in einer Bibliothek zu verwalten. Zusätzlich soll die Simulation der im Framework entwickelten Modelle gewährleistet sein und eine Schnittstelle zur Anwendung in Fahr simulatoren integriert werden. Zur Visualisierung der Hauptfunktionalitäten werden Use-Case-Diagramme erstellt, die eine Ähnlichkeit zu den Aktivitätsdiagrammen der Modellierungssprache *UML2* aufweisen (vgl. **Bild 3**). Sie beschreiben die Abläufe zur Sicherstellung der Funktionalität, aber geben keine konkreten Lösungsvorschläge vor. Somit können funktionale Anforderungen an das Framework abgeleitet werden.

4.3 Niederschreiben der Anforderungen

Anforderungs-Nr.:	25	Anforderungstyp:	2	Event/Use Case:	1
Beschreibung:	Beim Öffnen von bestehenden Fahrdynamikmodellen muss das Framework alle modellrelevanten Daten bereitstellen.				
Begründung:	Damit das Fahrdynamikmodell direkt verwendet bzw. simuliert werden kann, müssen neben dem Modell selbst alle relevanten Daten wie Parameter bereitgestellt werden.				
Erfüllungskriterium:	Nach dem Öffnen eines bestehenden Fahrdynamikmodells kann ohne weitere Aktionen eine Simulation durchgeführt werden.				
Priorität:	5				

Bild 4 Beispielhafte Snow Card

Um die zuvor ermittelten, potentiellen Anforderungen zu formalisieren und zu strukturieren, werden die von [7] entwickelten Snow Cards als Vorlage verwendet. Sie weisen einer Anforderung mehrere Attribute zu und tragen dazu bei, dass Anforderungen übersichtlich, nachvollziehbar und verfolgbar sind (vgl. **Bild 4**).

Der obere Bereich einer Snow Card dient einer Einordnung in den Kontext. Während die fortlaufende Nummerierung zur Strukturierung der späteren Anforderungsspezifikation beiträgt, erlauben die Definition des Anforderungstyps und des Use Cases eine Klassifizierung. Als Anforderungstyp kann hierbei zwischen einer Projektbeschränkung (1), einer funktionalen Anforderung (2) sowie einer nicht-funktionalen Anforderung (3) gewählt werden. Die Angabe des Use Cases stellt eine Beziehung zu den zuvor beschriebenen Use-Case-Diagrammen her und wird somit lediglich den funktionalen Anforderungen zugewiesen. Bei der in **Bild 4** dargestellten Anforderung handelt es sich somit um eine funktionale Anforderung, die dem Use Case 1 *Modifizieren von bestehenden Fahrdynamikmodellen* zugeordnet werden kann.

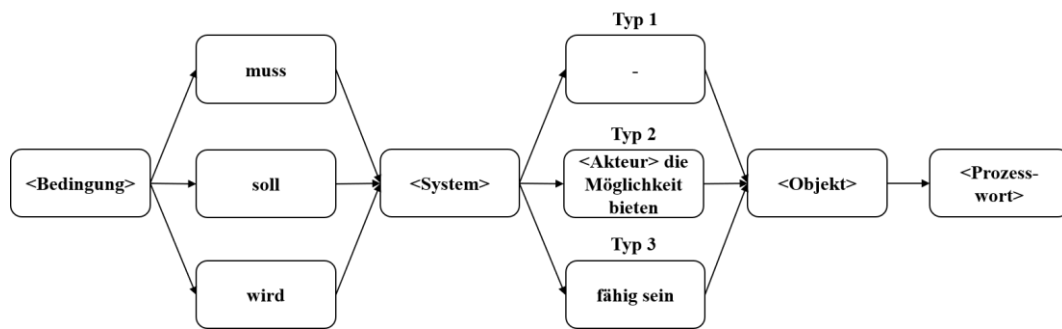


Bild 5 Anforderungsschablone, in Anlehnung an [6]

Ein zentraler Bestandteil der Snow Card ist die Beschreibung der Anforderung. Um hierbei den Qualitätsaspekten wie Formalität und Widerspruchsfreiheit gerecht zu werden, kommt die von [6] entwickelte Anforderungsschablone zum Einsatz. Sie legt die Struktur eines einzelnen Anforderungssatzes fest und basiert auf der Kombination von Satzbausteinen (vgl. **Bild 5**). Die Verbindlichkeit der Anforderung wird durch die Wahl des Wortes *muss*, *soll* oder *wird* festgelegt. Während *muss*-Anforderungen verbindlich sind und eine zentrale Rolle in der Formulierung von Anforderungen spielen, drückt die *soll*-Bedingung aus, dass die Berücksichtigung sinnvoll ist. *Wird*-Anforderungen verdeutlichen letztlich unverbindliche Wünsche der Stakeholder. Mit Hilfe der Schablone kann darüber hinaus zwischen drei Anforderungstypen differenziert werden. Der erste Typ beschreibt eigenständige Systemaktivitäten, die in der Regel mit den Use Cases des Systems zusammenhängen. Dahingegen drücken der zweite Typ eine Benutzerinteraktion und der dritte Typ eine Schnittstellenanforderung aus. Bezogen auf die in **Bild 4** dargestellte Anforderung liegt eine hohe Verbindlichkeit sowie eine eigenständige Systemaktivität vor. [6]

Jede formalisierte Anforderung wird zusätzlich mit einer ausführlichen Begründung versehen. Diese gibt Hintergründe zur Anforderung an und beschreibt die Notwendigkeit ihrer Berücksichtigung. Ferner wird ein Erfüllungskriterium festgehalten, welches nach der Umsetzung der Anforderung im Entwicklungsprozess als Testmittel und somit als Bindeglied zwischen der Anforderungsanalyse und -verfolgung dient. Zuletzt wird die Dringlichkeit der Umsetzung mit Hilfe einer Priorisierung dargestellt. Hier wird eine fünfstufige Skala verwendet, wobei die in **Bild 4** angegebene Priorität als Maximalwert zu verstehen ist.

4.4 Qualitätskontrolle und Anforderungsspezifikation

Die zuvor auf den Snow Cards niedergeschriebenen Anforderungen werden im letzten Schritt einer Qualitätskontrolle unterzogen, auf die an dieser Stelle nicht eingegangen wird. Im Anschluss werden die bewilligten Anforderungen in der Anforderungsspezifikation vereint. Hierzu wird das von [7] entwickelte Volere Requirements Specification Template verwendet. Es gliedert sich in die fünf Teile

- Projekttreiber,
- Projektbeschränkungen,

- funktionale Anforderungen,
- nicht-funktionale Anforderungen sowie
- Projektprobleme

und ist als vollständige Blaupause zur Beschreibung der Funktionalitäten und Fähigkeiten des Entwicklungsprojekts zu sehen. Darüber hinaus dient es aufgrund seines Aufbaus als Leitfaden während der Anforderungsanalyse und ebenso während des weiteren Entwicklungsprozesses. Zusammenfassend besteht die Anforderungsspezifikation aus 49 identifizierten und bewilligten Anforderungen. Diese unterteilen sich in sieben Projektbeschränkungen sowie 27 funktionale und 15 nicht-funktionale Anforderungen. Auf die detaillierte Ausarbeitung der Anforderungsspezifikation sei im Rahmen dieses Beitrags verzichtet.

5 Framework Vehicle Model

Nachdem im vorherigen Abschnitt der Prozess der Anforderungsanalyse dargelegt wurde, fokussiert sich der folgende Abschnitt auf die Ausarbeitung des Entwicklungsvorhabens mit dem Titel *Framework Vehicle Model*. In diesem Zusammenhang wird auch auf das Requirements Management eingegangen, welches sich unter anderem mit der Anforderungsverfolgung und dem Änderungsmanagement beschäftigt.

5.1 Gestaltung des Frameworks

Wie einleitend beschrieben und im Rahmen der Anforderungsanalyse spezifiziert soll eine eigenständige Lösung entwickelt werden, die eine möglichst hohe Flexibilität bietet. Daher soll auf kommerzielle Entwicklungswerkzeuge wie IPG Carmaker oder ADAMS/Car verzichtet werden. Der grundsätzliche Aufbau des Frameworks ist **Bild 6** zu entnehmen.

Kontextabgrenzung

Das Framework ist vollständig in die Softwareumgebung *MATLAB* integriert und speziell auf die Entwicklung von Fahrdynamikmodellen in *MATLAB/Simulink* ausgerichtet. Die Integration in *Simulink* ermöglicht eine Verwendung aller gängigen Werkzeuge zum Kompilieren von Modellen und bildet somit die Schnittstelle zu den Fahrsimulatoren des Lehrstuhls für Mechatronik.

Das Framework wird als Projekt in der Toolbox *Simulink Projects* angelegt, welches eine einheitliche, übersichtliche Entwicklungsplattform bietet und somit direkt zum Erfül-

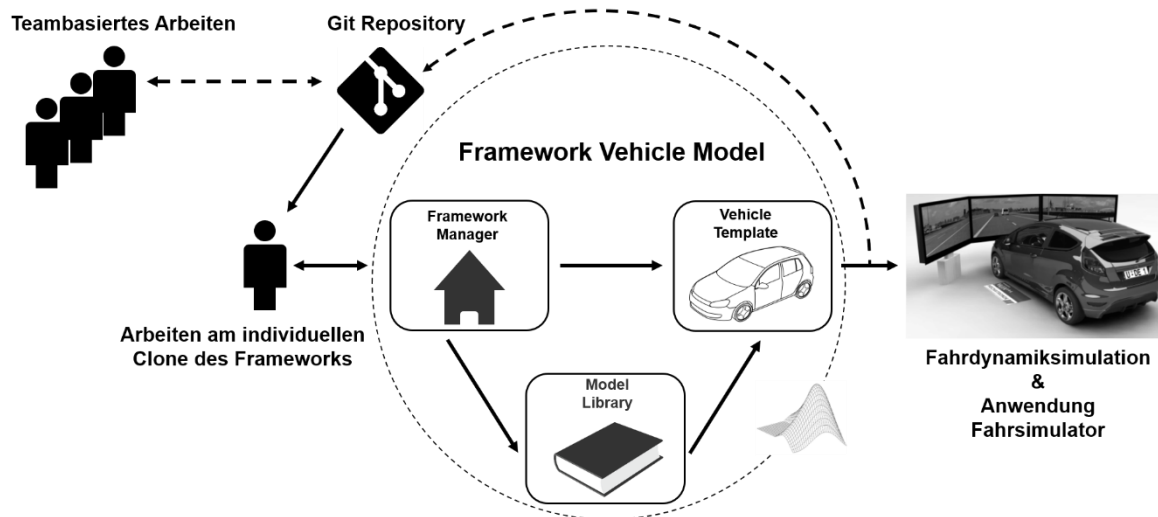


Bild 6 Entwicklung von Fahrdynamikmodellen mit Hilfe des Framework Vehicle Model

len einiger nicht-funktionaler Anforderungen wie Einheitlichkeit und Übersichtlichkeit beiträgt. Ferner unterstützt Simulink Projects die teambasierte Projektarbeit durch das Einbinden der Versionsverwaltungssoftware *Git*, dessen Verwendung in den Projektbeschränkungen gefordert wird. Durch das Ablegen des Git-Repository in der Projektmanagementsoftware *Redmine* wird einem skalierbarem Personenkreis Zugriff auf das Framework gewährt. Jeder Nutzer kopiert den aktuellen Arbeitsstand des Frameworks sowie der integrierten Modelle als Clone aus dem Git-Repository, um anschließend selbstständige Änderungen vorzunehmen. Ein hierbei entstandenes Fahrdynamikmodell kann anschließend auf einen Fahrsimulator übertragen und simuliert werden. Zuletzt können Änderungen zurück in das Git-Repository übertragen werden (vgl. **Bild 6**).

Entwurfsentscheidungen

Die grundlegenden Entwurfsentscheidungen betreffen vor allem die Gestaltung von Fahrdynamikmodellen in MATLAB/Simulink, die mit Hilfe der Interviews und des Beobachtens bestehender Systeme im Rahmen der Anforderungsidentifikation ermittelt wurden. Somit werden hier vor allem Projektbeschränkungen und nicht-funktionale Anforderungen bedient.

Um dem Wunsch nach einer Komponentisierung der Simulationsmodelle gerecht zu werden, wird in Anlehnung an [9] eine Unterteilung in charakteristische Subsysteme (Steering Systems, Braking Systems, Powertrains, Chassis, Suspension Systems, Tires) vorgenommen. Zusätzlich werden zwei weitere Subsysteme verwendet, um zum einen die Fahrerschnittstelle zu bedienen (Information Systems) und zum anderen die Online-Visualisierung sowie den Export von Fahrdynamikgrößen zu ermöglichen (Visualization Systems).

Zur Kopplung der Modellkomponenten wird eine einheitliche Schnittstelle verwendet, die in Anlehnung an das Bondgraph-Konzept den Fluss von Leistung mit Hilfe der Variablen $Effort\ e(t)$ und $Flow\ f(t)$ beschreibt [10]. Als Entscheidungsgrundlage dienen die Ausführungen in [3], in denen eine Erweiterung der Leistungsvariablen um eine Informationsgröße vorgenommen wird. Diese hilft bei der

Kopplung von Modellkomponenten, die keinen klaren Leistungsfluss aufweisen.

Das Konzept zur Signalverwaltung ermöglicht maßgeblich die Erfüllung der nicht-funktionalen Anforderungen bezüglich der Übersichtlichkeit der im Framework erstellten Fahrdynamikmodelle. Hier kommt es zur Verwendung von Busobjekten in Kombination mit einem globalen Datenspeicher. Ein Busobjekt beinhaltet die Definition der Eigenschaften wie Name, Datentyp oder Einheit des Signals und wird in Anlehnung an das zuvor erläuterte Schnittstellenkonzept gebildet. Der Datentyp kann hierbei wiederum ein Busobjekt sein, wodurch eine Verschachtelung der Signale vorgenommen werden kann. Während die Verwendung von Busobjekten die Wiederverwendbarkeit, die Austauschbarkeit und die Flexibilität von Modellen erhöht, führt der globale Datenspeicher zu einer Reduzierung der Signallinien im Simulinkmodell. Er wird mittels der Data-Store-Memory-Funktion realisiert und ermöglicht die vollständige Entkopplung von Modellkomponenten, welche wiederum die Übersichtlichkeit und Austauschbarkeit der Fahrdynamikmodelle und deren Komponenten erhöht. Letztlich dienen weitere, im Framework definierte Richtlinien wie die Vorgabe einer Dateistruktur sowie die Umsetzung einer Namens- und Farbkonvention zur Erfüllung der nicht-funktionalen Anforderungen.

Bestandteile des Frameworks

Das Framework Vehicle Model setzt sich aus den Komponenten

- Framework Manager,
- Model Library und
- Vehicle Template

zusammen. Die schlanke Struktur dient hierbei der Übersichtlichkeit und Anwendungsfreundlichkeit (vgl. **Bild 6**). Der Framework Manager kann als übergeordnetes Verwaltungsprogramm verstanden werden und stellt eine umfassende Anwenderschnittstelle dar. Diese ermöglicht eine Steuerung des Frameworks ohne genaue Kenntnis über die Abwicklung aller Funktionen, initiiert sämtliche Haupt-

funktionalitäten und dient darüber hinaus als Rückfallebene im Fehlerfall. Aus Gründen der Anwendungsfreundlichkeit ist der Framework Manager als grafische Benutzeroberfläche realisiert.

Die Model Library verwaltet die im System eingepflegten Modellkomponenten des Fahrzeugs sowie der Informationssysteme zur Realisierung der Fahrerschnittstelle. Sie orientiert sich im Aufbau und der Funktion an der von Simulink bereitgestellten Bibliothek *Simulink Library Browser* und ist in diese integriert. Das Hinzufügen von Modellkomponenten in ein Fahrdynamikmodell geschieht über eine Drag-and-Drop-Operation.

Zuletzt dient das Vehicle Template als Basis für jedes im Framework entwickelte Fahrdynamikmodell und ist somit zentraler Bestandteil der Architektur. Es handelt sich hierbei um ein Simulinkmodell, welches auf der obersten Ebene aus den im vorherigen Abschnitt beschriebenen Subsystemen besteht. Zusätzlich ist hier auf oberster Ebene der Block für den globalen Datenspeicher mit dem Namen *GlobalBus* angeordnet. In die einzelnen Subsysteme lassen sich wiederum die entsprechenden Modellkomponenten aus der Model Library einfügen.

5.2 Requirements Management

Jedes Projekt unterliegt veränderlichen Anforderungen. Hierbei wird hinsichtlich des zeitlichen Auftretens in

- Anforderungen, die zu Projektbeginn gestellt werden,
- Anforderungen, die im Laufe des Projekts gestellt und als Änderungswünsche bezeichnet werden, sowie
- Anforderungen, die nachträglich gestellt werden und neue Anforderungen, Fehlermeldungen oder Verbesserungsvorschläge darstellen,

differenziert [4]. Im Hinblick auf die Entwicklung des hier vorgestellten Frameworks sei angemerkt, dass im Projektverlauf stetig Änderungswünsche geäußert und nachträgliche Anforderungen definiert wurden. Aufgrund der übersichtlichen Struktur der Anforderungsspezifikation und der in ihr enthaltenen Snow Cards konnten diese Anforderungen ebenfalls integriert werden. Die konkrete Umsetzung wurde durch die bewusst modular gewählte Architektur des Frameworks realisiert. Ferner sei an dieser Stelle darauf hingewiesen, dass das in Abschnitt 3 vorgestellte V-Modell im realen Projektverlauf mehrfach durchlaufen wurde, um alle Änderungswünsche und nachträglich gestellten Anforderungen zu berücksichtigen.

Neben der Berücksichtigung von veränderlichen Anforderungen ist die Anforderungsverfolgung eine weitere Aufgabe des Requirements Managements. Sie liefert Informationen über den Einfluss von Anforderungen, deren Änderungen sowie den aktuellen Entwicklungsstand [8]. Dies wird hier ebenfalls mit Hilfe der Snow Cards realisiert, indem sowohl Verknüpfungen zu den in der Anforderungsanalyse identifizierten Hauptfunktionalitäten hergestellt werden können als auch jede Anforderung mit einem Erfüllungskriterium versehen ist. Letzteres ermöglicht während der im V-Modell vorgesehenen Testphasen eine Bewertung des Entwicklungsfortschritts und am Projektende

die Evaluierung der Umsetzung des Entwicklungsvorhabens.

In diesem Zusammenhang wird jede in der Anforderungsspezifikation festgehaltene, funktionale und nicht-funktionale Anforderung am Projektende getestet und bewertet. Von den 27 funktionalen Anforderungen werden lediglich drei nicht vollständig erfüllt. Dies betrifft insbesondere den Automatisierungsgrad der im Framework implementierten Funktionen. Das heißt, dass die Anforderung bereits manuell bewerkstelligt werden kann, jedoch durch die Formulierung der Anforderung eine automatisierte Aktion des Frameworks gewünscht wird. Diese gilt es in späteren Revisionen einzupflegen. Einen wichtigen Beitrag zur Bewertung der Qualität des Frameworks liefern die nicht-funktionalen Anforderungen, die allerdings subjektiven Einflüssen unterliegen. Hierzu zählen die Anwendungsfreundlichkeit, Einheitlichkeit und Übersichtlichkeit des Frameworks und der in ihm entwickelten Fahrdynamikmodelle. Diese Faktoren werden durch zahlreiche Maßnahmen wie die Integration der bereits beschriebenen Konventionen berücksichtigt. Darüber hinaus dienen die schlanke Struktur des Frameworks mit drei Bestandteilen sowie insbesondere die Umsetzung des Framework Managers in einer grafischen Benutzeroberfläche der Anwendungsfreundlichkeit. Letztlich unterstützt die Anfertigung einer umfassenden Dokumentation beim Erlernen der Funktionen des Frameworks. Dies schließt eine Schnellklick-Anleitung sowie einen Leitfaden zur Einhaltung der Namens- und Farbkonvention ein und wirkt sich positiv auf die Akzeptanz der Nutzer aus.

6 Zusammenfassung und Ausblick

Zur Forschung und Entwicklung innovativer Technologien wird sowohl in der Automobilindustrie als auch in anderen Industriezweigen zunehmend auf Simulationstechniken zurückgegriffen. Vor diesem Hintergrund wurde im vorliegenden Beitrag die Entwicklung eines Software-Frameworks zur Erstellung von Fahrdynamikmodellen beschrieben. Während das hier entwickelte Framework Vehicle Model speziell auf Fahrdynamikmodelle zugeschnitten ist, sind die Schwerpunkte dieses Beitrags, die zugrundeliegende Entwicklungsmethodik sowie die Analyse der Benutzer- und Softwareanforderungen, auf andere Domänen uneingeschränkt übertragbar.

Das Entwicklungsvorhaben orientierte sich methodisch am V-Modell, welches mit Hilfe von Phasenreviews um iterative Ansätze erweitert wurde. Die Gründe für diese Entscheidung waren, neben der allgemeinen Akzeptanz gegenüber der Methodik, der vergleichsweise geringe Projektumfang sowie der kleine Entwicklerkreis. Während das klassische V-Modell eine einmalige Bearbeitung aller Phasen vorsieht, zeigte der reale Projektverlauf, dass das gesamte Entwicklungsvorhaben nicht in einem Durchlauf zu bewerkstelligen war. Somit wurden die einzelnen Phasen des V-Modells mehrfach durchlaufen, um Änderungswünsche zu berücksichtigen.

Aufgrund der Struktur des hier gewählten Vorgehensmodells, so dass Fehler aus frühen Entwicklungsphasen spät aufgedeckt werden, kommt einer ausführlichen Anforderung

rungsanalyse eine besondere Bedeutung zu. Hier wurde der Volere-Prozess eingesetzt, der sich in die Anforderungsidentifikation, das Niederschreiben der Anforderungen sowie einer abschließenden Qualitätskontrolle unterteilt. Ferner bietet das Volere Requirements Specification Template einen Leitfaden zur strukturierten Erstellung der Anforderungsspezifikation. In diesem Zusammenhang wurden die Projektbeschränkungen und funktionalen sowie nicht-funktionalen Anforderungen an die Entwicklung des Frameworks identifiziert. Als Hauptbestandteil der Anforderungsspezifikation dienten Snow Cards. Diese erlauben durch die Zuordnung mehrerer Attribute eine Klassifizierung der Anforderung und bieten Vorteile bei der Anforderungsverfolgung während des Entwicklungsprozesses. Darüber hinaus ist jede Snow Card mit einer Anforderungsbeschreibung, einer ausführlichen Begründung sowie einem Erfüllungskriterium versehen, so dass eine Anforderung vollständig spezifiziert, nachvollziehbar und prüfbar ist. Somit konnten alle Qualitätsaspekte berücksichtigt werden, die eine ausführliche Anforderungsanalyse aufweisen soll. Im Rahmen der hier durchgeführten Analyse wurden 49 Anforderungen identifiziert.

Das Framework Vehicle Model wurde unter Berücksichtigung der ermittelten Anforderungen in MATLAB/Simulink umgesetzt und setzt sich aus drei Bestandteilen zusammen, die um ein umfassendes Regelwerk ergänzt werden. Hierbei verfolgt das Framework das Ziel, eine modulare Architektur zur komponentenbasierten Erstellung von Fahrdynamikmodellen zu bieten. Dieser Ansatz sowie die im vorliegenden Beitrag vorgestellte Entwicklungsmethodik und Anforderungsanalyse lassen sich auf viele weitere technische Domänen übertragen.

Im Fokus der weiteren Arbeiten steht die Implementierung weiterer Anwenderwünsche und die Optimierung der Funktionsabläufe. Hierzu zählt insbesondere die fortschreitende Automatisierung von Funktionen, um den Aufwand bei der Verwendung des Frameworks zu reduzieren und somit die Akzeptanz der Nutzer weiter zu erhöhen. Darüber hinaus werden hierdurch mögliche Anwendungsfehler vermieden. Ein weiterer Schwerpunkt zukünftiger Arbeiten ist die Einbeziehung zusätzlicher Fahrzeugfunktionen wie Fahrerassistenzsysteme, um die Nutzbarkeit des Frameworks weiter zu erhöhen. Während dieser kontinuierlichen Verbesserungsprozesse erfolgt die Integration des Frameworks in die täglichen Entwicklungsarbeiten der Stakeholder. In diesem Zusammenhang ist die Berücksichtigung und Umsetzung der im Framework vorgesehenen Versionsverwaltung mit Hilfe der Software *Git* von besonderer Bedeutung. Hierdurch soll das Framework Vehicle Model als Entwicklungswerkzeug am Lehrstuhl für Mechatronik der Universität Duisburg-Essen etabliert werden.

7 Literatur

- [1] D. Schramm, B. Hesse, N. Maas und M. Unterreiner: *Fahrzeugtechnik: Technische Grundlagen aktueller und zukünftiger Kraftfahrzeuge*. Berlin, Boston: Walter de Gruyter, 2017.
- [2] A. P. Rousseau et al.: *Electric Drive Vehicle Development and Evaluation Using System Simulation*. In: Proceedings of the 19th World Congress, Kapstadt, 2014, S. 7886–7891.
- [3] S. Halbach, P. Sharer, S. Pagerit, C. Folkerts und A. P. Rousseau: *Model Architecture, Methods, and Interfaces for Efficient Math-Based Design and Simulation of Automotive Control Systems*. In: SAE Technical Paper, 2010.
- [4] J. Schäuffele und T. Zurawka: *Automotive Software Engineering: Grundlagen, Prozesse, Methoden und Werkzeuge effizient einsetzen*. 6. Aufl., Wiesbaden: Springer Fachmedien, 2016.
- [5] T. Pfeifer und R. Schmitt: *Masing Handbuch Qualitätsmanagement*. 6. Aufl., München [u.a.]: Hanser, 2014.
- [6] C. Rupp: *Requirements-Engineering und -Management: Aus der Praxis von klassisch bis agil*. 6. Aufl., München: Hanser, 2014.
- [7] S. Robertson und J. Robertson: *Mastering the requirements process: Getting requirements right*. 3. Aufl., Upper Saddle River, NJ [u.a.]: Addison-Wesley, 2013.
- [8] C. Ebert: *Systematisches Requirements Engineering: Anforderungen ermitteln, dokumentieren, analysieren und verwalten*. 5. Aufl., Heidelberg: dpunkt-Verlag, 2014.
- [9] D. Schramm, M. Hiller und R. Bardini: *Modellbildung und Simulation der Dynamik von Kraftfahrzeugen*. 3. Aufl., Berlin, Heidelberg: Springer-Verlag, 2018.
- [10] D. Karnopp, D. L. Margolis und R. C. Rosenberg: *System dynamics: Modeling simulation and control of mechatronic systems*. 5. Aufl., Hoboken, N.J.: Wiley, 2012.

Ein Hardware-in-the-Loop-Prüfstand zur Verifikation und Validierung von Matrixscheinwerfern

Mirko Waldner, M.Sc., TU Dortmund, Institut für Roboterforschung, Lehrstuhl für Regelungssystemtechnik, 44221 Dortmund, Deutschland, mirko.waldner@tu-dortmund.de

Maximilian Krämer, M.Sc., TU Dortmund, Institut für Roboterforschung, Lehrstuhl für Regelungssystemtechnik, 44221 Dortmund, Deutschland, maximilian.kraemer@tu-dortmund.de

Univ.-Prof. Dr.-Ing. Prof. h.c. Dr. h.c. Torsten Bertram, TU Dortmund, Institut für Roboterforschung, Lehrstuhl für Regelungssystemtechnik, 44221 Dortmund, Deutschland, torsten.bertram@tu-dortmund.de

Kurzfassung

Moderne Matrix-Scheinwerfersysteme bestehen aus einer Vielzahl verwendeter Technologien [1, 2] und Modulen, mit denen die Lichtverteilung an die aktuelle Umgebungssituation angepasst wird. Die Kollaboration verschiedener Module realisiert bspw. Funktionen wie adaptive Frontscheinwerfersysteme (AFS), blendfreies Fernlicht [3] und die Projektion von Symbolen [4] auf die Straße. Die Funktionalität von Matrixscheinwerfern wird im Entwicklungsprozess laufend über reale Testfahrten verifiziert und validiert, die nur bei entsprechender Wetterlage und Tageszeit, wie zum Beispiel Trockenheit und Dunkelheit, durchgeführt werden können. Eine Ergänzung zu realen Testfahrten ist der in diesem Beitrag vorgestellte Hardware-in-the-Loop-Prüfstand (HiLP), mit dem Verifikationen und Validierungen zu jeder Zeit im Labor oder Lichtkanal reproduzierbar durchführbar sind. Dies verbessert das präzise Einstellen der Designparameter und das Auffinden von Fehlern in immer komplexer werdenden mechatronischen Systemen.

Der Prüfstand ermöglicht zwei Evaluierungsszenarien, zum einen die direkte visuelle Bewertung der realen Lichtverteilung unter Einfluss der Fahrdynamik und zum anderen die Integration der realen Lichtverteilung in ein virtuelles Prüfzenario. Die Systemarchitektur und die Informationsflüsse sind in Abbildung 1 dargestellt. Matrixscheinwerfer mit 84 Pixeln der Firma HELLA™ GmbH & Co. KGaA. dient als Demonstrator, an dem die Eignung und Güte des HiLP-Ansatzes als Entwicklungsmethode erforscht wird.

Die Fahrdynamik des Egofahrzeuges, welches das zu untersuchende Matrixscheinwerferpaar trägt, wird entweder über ein Mehrkörpermodell simuliert oder aus gemessenen Fahrdynamikdaten einer Testfahrt rekonstruiert. Die Zustandsgrößen des Fahrzeuges und eine virtuelle Umgebung, welche unter anderem aus einer Wetter- und Verkehrssimulation besteht, bilden das Fahrzeug-Umfeld-Modell (Abb. 1.a). Das Modell generiert die Scheinwerfertrajektorien als Vorgabe für die mechanische Aktuierung (Abb. 1.c) des realen Scheinwerfers. Mithilfe eines seriellen Roboterarms werden die mechanischen Anforderungen, der Aufbau und geeignete Regelungskonzepte für eine genaue und zuverlässige Nachbildung der Fahrdynamik erforscht. Ein virtuelles Sensorsystem erfasst die relevanten virtuellen Verkehrsobjekte und Zustände des Egofahrzeuges und liefert damit die Eingangsgrößen für die zu testenden Lichtfunktionen (Abb. 1.b). Die Ansteuerung der einzelnen Module gemäß vorgegebener Steuerparameter adaptiert die reale Lichtverteilung an die virtuelle Verkehrssituation.

Für das zweite Evaluierungsszenario, in dem die reale Lichtverteilung digitalisiert wird, verbleibt das Bewegungssystem in Ruhe. Der Matrixscheinwerfer scheint auf eine Projektionsfläche (Abb. 1.d), auf der die Lichtverteilung von einem Kamerasystem optisch erfasst wird. Anschließend errechnet die Bildverarbeitung (Abb. 1.e) eine Helligkeitsverteilung (Abb. 1.f). Nach einer Rauschunterdrückung werden die Pixelwerte durch Vergleich mit einer gemessenen Lichtstärkeverteilung (LSV) optimiert und abschließend das Bild für die Transformation in eine virtuelle Lichtverteilung vorbereitet. Damit die Bildverarbeitungspipeline nicht nur die Lichtverteilungen des 84 LED Scheinwerfers realitätsnah digitalisiert, sondern auch auf möglichst alle aktuellen und zukünftigen Technologien [1, 2] generalisierbar ist, werden offline Lichtstärkenmessungen als Grundlage für die Justierung verwendet. Die automatische Justierung minimiert den Messfehler der Lichtstärke durch die optimale Adaption der Kameraparameter, wie z.B. Belichtungszeit und Öffnungsweite, und Verarbeitungsfilter, wie z.B. Gammakorrektur und Rauschunterdrückung, an die offline Lichtverteilung. Die Auswahl, Entwicklung und Parametrisierung von Filteralgorithmen zur Verbesserung der realitätsnahen Transformation bei optimaler Justierung wird hierbei untersucht. Die realitätsnahe Transformation umfasst nicht nur den allgemeinen Helligkeitsverlauf, sondern insbesondere auch die bläulichen chromatischen Aberrationen und Inhomogenitäten bei der Überlappung der Ausleuchtungsbereiche einzelner LED's. Eine Komponente des virtuellen Fahrzeug-Umfeld-Modells (Abb. 1.a) ist eine echtzeitfähige Lichtsimulation mit einer Standardarchitektur [5, 6], in der die Helligkeitsverteilung des realen Matrixscheinwerfers unter Verwendung von projiziertem Textur Mapping [7] dargestellt wird. In der virtuellen Welt wird der Einfluss der Bildverarbeitung auf die subjektive qualitative Realitätsnähe der Transformation und den Lichteindruck erforscht.

Dieses Vorhaben wird aus Mitteln des Europäischen Fonds für regionale Entwicklung (EFRE) unter dem Förderkennzeichen EFRE-0800850 gefördert.

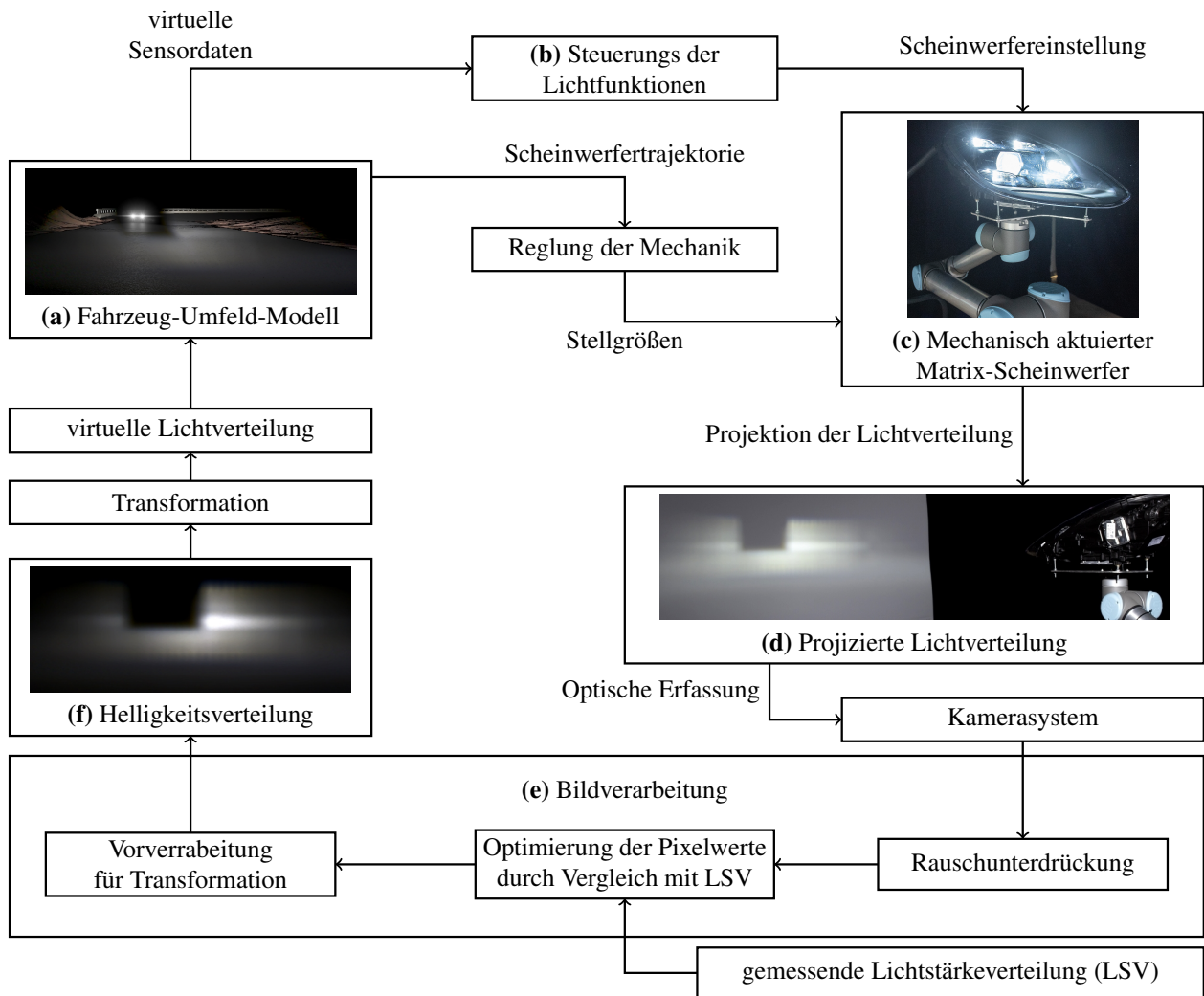


Abbildung 1 Gesamtübersicht über den Hardware-in-the-Loop-Prüfstand.

Literatur

- [1] Moisel, J.: *Requirements for future high resolution ADB modules*. In: 11th International Symposium on Automotive Lighting – ISAL 2015, Darmstadt, S. 161-170.
- [2] Duhume, D. und Fischer, B.: *Next Generation LCD Module*. In: 12th International Symposium on Automotive Lighting – ISAL 2017, Darmstadt, S. 11-18.
- [3] Hummel, B.: *Blendfreies LED-Fernlicht*. Audi Dissertationsreihe, Cuvillier Verlag, 2010.
- [4] Kubitz, B. und Wilks, C.: *Digitales Licht als Unterstützung für den Fahrer*. ATZ - Automobiltechnische Zeitschrift 4 2018, S. 56-59.
- [5] Berssenbrügge, J., Gausemeier, J., Grafe, M., Matysczok, C. und Pöhland, K.: *Real-time Representation of Complex Lighting Data in a Nightdrive Simulation*. EGVE '03 Proceedings of the Workshop on Virtual Environments 2003.
- [6] Lecocq, Pascal, M. Kelada, J und Kemeny, Andras: *Interactive Headlight Simulation*. 2004 <http://www.experts.renault.com/DSC99/papers/P25.PDF> (Abgerufen 9.2018).
- [7] Segal, M., Korobkin, C., van Widenfelt, R., Foran, J. und Haeberli, P.: *Fast Shadows and Lighting Effects Using Texture Mapping*. SIGGRAPH '92 Proceedings of the 19th Annual Conference on Computer Graphics and Interactive Techniques, 1992.

Fünfte IFToMM D-A-CH Konferenz 2019
26./27. Februar 2019, RWTH Aachen

In: IFToMM D-A-CH Konferenz

Dieser Text wird über DuEPublico, dem Dokumenten- und Publikationsserver der Universität Duisburg-Essen, zur Verfügung gestellt.

Die hier veröffentlichte Version der E-Publikation kann von einer eventuell ebenfalls veröffentlichten Verlagsversion abweichen.

DOI: <https://doi.org/10.17185/duepublico/47897>

URN: <urn:nbn:de:hbz:464-20190222-142110-1>

Link: <https://duepublico.uni-duisburg-essen.de:443/servlets/DocumentServlet?id=47897>

Lizenz:

Sofern nicht im Inhalt ausdrücklich anders gekennzeichnet, liegen alle Nutzungsrechte bei den Urhebern bzw. Herausgebern. Nutzung - ausgenommen anwendbare Schrankenregelungen des Urheberrechts - nur mit deren Genehmigung.