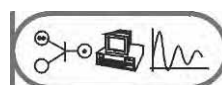


Virtueller Workshop ASIM STS/GMMS & EDU 2021 TAGUNGSBAND

11.-12. März 2021

Images: DLR, CC-BY 3.0



ARGESIM Reports

Published by **ARGESIM** and **ASIM**, Arbeitsgemeinschaft Simulation,
Fachausschuss 4.5 der GI

Series Editor:

Felix Breitenecker (ARGESIM / ASIM)
Math. Modelling and Simulation Group, TU Wien
Wiedner Hauptstrasse 8 - 10, A - 1040 Vienna
Tel: +43-1-58801-10115, Fax: +43-1-58801-910115
Email: Felix.Breitenecker@tuwien.ac.at

ARGESIM Report AR45 ASIM Mitteilung AM 175

Titel: Tagungsband
Virtueller ASIM Workshop 2021
Simulation Technischer Systeme
Grundlagen und Methoden in Modellbildung und Simulation
Edukation und Simulation

Herausgeber: Xiaobo Liu-Henke
Umut Durak

ISBN Online: 978-3-901608-69-8
DOI: 10.11128/arep.45

Das Werk ist urheberrechtlich geschützt. Die dadurch begründeten Rechte, insbesondere die der Übersetzung, des Nachdrucks, der Entnahme von Abbildungen, der Funksendung, der Wiedergabe auf photomechanischem oder ähnlichem Weg und der Speicherung in Datenverarbeitungsanlagen bleiben, auch bei nur auszugsweiser Verwertung, vorbehalten.

© by ARGESIM / ASIM, Wien, 2021

ARGE Simulation News (ARGESIM)
c/o F. Breitenecker, Math. Modelling and Simulation Group, TU Wien
Wiedner Hauptstrasse 8 - 10, A - 1040 Vienna
Tel: +43-1-58801-10115, Fax: +43-1-58801-910115
Email: info@argesim.org; WWW: www.argesim.org

Tagungsband

Virtueller ASIM Workshop 2021
Simulation Technischer Systeme
Grundlagen und Methoden in Modellbildung
und Simulation
Edukation und Simulation

11. bis 12. März 2021

Zusammenfassung der Beiträge
Xiaobo Liu-Henke(Hrsg.)
Umut Durak (Hrsg.)



Arbeitsgemeinschaft Simulation ASIM in der Gesellschaft für Informatik GI

Tagungsleitung:

Prof. Dr.-Ing. Xiaobo Liu-Henke (Ostfalia Hochschule)

Programmkomitee:

Robert Buchta, Volkswagen

Walter Commerell, Hochschule Ulm

Umut Durak, DLR Braunschweig

Leo Gall, LTX Simulation GmbH, München

Andreas Körner, TU Wien

Xiaobo Liu-Henke, Ostfalia HAW

Daniel Lückerrath, Fraunhofer IAIS

Heinz-Theo Mammen, Hella KGaA Hueck & Co., Lippstadt

Thorsten Pawletta, Hochschule Wismar

Nikolas Popper, dwh GmbH, Wien

Michael Striebel, HTWG Konstanz

Siegfried Wassertheurer, AIT

**Virtueller ASIM Workshop 2021
Simulation Technischer Systeme
Grundlagen und Methoden in
Modellbildung und Simulation
Edukation und Simulation
Tagungsband**

INHALT

INHALT

Vorwort	1
 Modellbasierte Mechatronikentwicklung	
Automatisierte Umgebungsmodellierung auf Basis von Sensormessdaten zur Erprobung intelligenter Fahrzeugfunktionen mittels driving-simulator-in-the-Loop-Simulationen <i>Marian Göllner, Sven Jacobitz, Sören Scherler, Jie Zhang, Or Aviv Yarom, Xiaobo Liu-Henke</i>	3
Presentation of a Model Integration Concept for Real Time Simulation (Kurzfassung) <i>Oliver Ellis</i>	13
Mobiles Labor auf Open-Source-Basis für die Ausbildung mechatronischer Antriebssysteme <i>Rolf Roskam</i>	15
Qualification Considerations for Simulations in Avionics Software Engineering <i>Mohamad Ibrahim, Umut Durak</i>	23
 Simulation in der Elektronikentwicklung	
SyNumSeS: a Python package for numerical simulation of semiconductor devices <i>Peter Abele, Michael Schäfer</i>	31
Orthotropic non-hysteretic permeability model as a lookup-table for FEA <i>Robert Courant, Jürgen Maas</i>	37
Simulation des Einflusses von Oberschwingungen auf die Spannungsqualität und Übertragungsverluste <i>Benjamin Jacobsen</i>	45
Modeling of mechatronics system on example on window regulator: conceptions and troubles <i>Sergey Petkun</i>	49
 Mathematische Verfahren in Modellbildung und Simulation	
Rechenzeitoptimierte Modellierung nicht-linearer physikalischer Prozesse mit Surrogate Modellen zur Anwendung in echtzeitfähigen Optimierungsverfahren <i>Marian Göllner, Or Aviv Yarom, Jannis Fritz, Xiaobo Liu-Henke</i>	57
Simulation Study on Methods for Modeling Elastic Collisions <i>Milena Sipovac, Andreas Körner</i>	65
Simulationsmodell zur Generierung von Trainingsdaten für ML-gestützte Ansätze der Zustandsüberwachung <i>Dennis Jess, Martin Strube</i>	69

Grundlagen und Methoden in Modellbildung und Simulation

Systems Engineering als Basis für konstruktionsbegleitende Zusammenarbeit	
<i>Eva Russwurm, Florian Faltus, Joerg Franke</i>	75
Entwicklung einer Simulationsumgebung für die automatisierte Modellkonfiguration zur Auslegung und Absicherung KI-basierter Fahrfunktionen	
<i>Or Aviv Yarom, Xiaobo Liu-Henke</i>	85
Schnittstellenimplementierung zum Aufbau einer numerischen Prozesskette in der Eisengussindustrie	
<i>Wolfgang Schlüter, Nazar Adamchuk</i>	93

Tutorials

Tutorial: Variantenmodellierung und automatisierte Simulationsexperimente	
<i>Thorsten Pawletta, Hendrik Folkerts, Christina Deatcu</i>	95

Simulation cyber-physischer Systeme

Real-time detection and classification for a 360°-camera using a YOLO algorithm	
<i>Tetiana Lavrenko, Ayman Ahmed, Vladimir Prokopenko, Thomas Walter, Hubert Mantz..</i>	97
Cloudbasierte V2X Kommunikation zur dynamischen Zielführung im vernetzten Verkehr	
<i>Sören Scherler, Lars Rüdiger Kelm, Sven Jacobitz, Florian Pramme, Gert Bikker</i>	103
<i>Xiaobo Liu-Henke</i>	
Simulation Scenarios for Testing Airborne Cyber-Physical Systems	
<i>Siddhartha Gupta, Umut Durak, Sven Hartmann</i>	113
Modellbasierte Entwicklung des Antriebsmoduls für vernetzte fahrerlose Transportfahrzeuge in einem cyber-physischen Labortestfeld	
<i>Sven Jacobitz, Marian Göllner, Jie Zhang, Xiaobo Liu-Henke</i>	117

Simulation technischer Systeme

Simulation eines mobilen Manipulators mit Gazebo	
<i>Kai Mario Kriegel, Dagmar Meyer, Steffen Ring, Christian Klöppelt</i>	125
Virtueller Prüfstand zur simulationsgestützten Bewertung von cyber-physischen Industrie-4.0-Lösungen	
<i>Jie Zhang, Xiaobo Liu-Henke</i>	133
Simulation Model of a Torque-Limiting Clutch with Adjustable Design Parameters to Investigate the Release Behavior	
<i>Michael Steck, Daniel Paland, Thomas Gwosch, Sven Matthiesen</i>	141
Modellierung des Betriebsverhaltens eines Diesel-Industriemotors als Stromerzeuger	
<i>Daniel Jörss, Bert Henrik Herrmann, Christian Fink</i>	149

Angewandte Simulationen

Simulation von Energiekonzepten in mittelständischen Gewerbegebieten

Maximilian Stange, Tom Wolf, Benjamin Jacobsen..... 157

Ein Fitnesscoach für Menschen mit Querschnittslähmung

Kilian Nickel 159

ARCH – Saving Cultural Heritage from Climate-related and Other Hazards

Daniel Lückerrath, Katharina Milde, Oliver Ullrich..... 161

Kohärenzmodell gesunder Online-Arbeit mit exemplarischer Ableitung von Lernregeln für die Unterstützung effektiver virtueller Lernaktivitäten

Marianne Schmolke..... 165

Werkstatt zu digitalen Online-Prüfungen

Online Prüfungen mit STACK Aufgaben

Gerda Fiedler, Birgit Gottschlich-Müller, Karin Melcher..... 173

Heuristische Untersuchung von Online-Prüfungen an der TU Wien

Corinna Modiz, Alexander Edthofer, Andreas Körner..... 179

Digitale Aufgaben und Prüfungen in Mathematik – ein Erfahrungsbericht

Karin Landenfeld, Jonas Priebe, Niels Gandraß..... 185

**Virtueller ASIM Workshop 2021
Simulation Technischer Systeme
Grundlagen und Methoden in
Modellbildung und Simulation
Edukation und Simulation
Tagungsband**

Vorwort

Simulationstechnologie ist in vielen Bereichen und insbesondere im Entwicklungsprozess technischer Systeme fest verankert. Der jährliche Workshop der ASIM/GI-Fachgruppen STS (Simulation Technischer Systeme), GMMS (Grundlagen und Methoden in Modellbildung und Simulation) bietet eine breite Plattform, um sowohl den aktuellen technologischen Stand als auch zukünftige Chancen in Theorie und Praxis zu beleuchten. Wir laden Sie herzlich ein, mit Experten aus Industrie und Wissenschaft zu diskutieren.

Dank vieler engagierter Autorinnen und Autoren wurden hochqualitative Beiträge mit aktuellen Forschungsergebnissen eingereicht. Wir haben uns über wachsende, neue Themengebiete sehr gefreut – neben den traditionellen Inhalten des Workshops sind auch die neuen Themen „Künstliche Intelligenz“ und „Maschinelles Lernen“ hinzugekommen.

Der wissenschaftliche Austausch kann auch im Jahr 2021 aufgrund der Corona-Pandemie nur virtuell erfolgen. Daher findet der diesjährige Workshop online – aber ebenso mit Möglichkeiten zu intensiver Diskussion und Vernetzung statt. Gerade in dieser Pandemiezeit sind Methoden zur virtuellen Online-Lehre unabdingbar, daher haben wir die Themengebiete um die digitale Online-Lehre als einen festen Bestandteil des diesjährigen Workshops erweitert.

Der vorliegende Tagungsband enthält die Beiträge des Workshops, die insgesamt folgende Themenschwerpunkte behandeln:

- Modellbasierte Mechatronikentwicklung
- Simulation in der Elektronikentwicklung
- Mathematische Verfahren in Modellbildung und Simulation
- Grundlagen und Methoden in Modellbildung und Simulation
- Simulation Cyber-Physischer Systeme
- Simulation Technischer Systeme
- Angewandte Simulationen
- Werkstatt zu Digitalen Online-Prüfungen

Für die Unterstützung und Mithilfe bei der Organisation des diesjährigen virtuellen Workshops möchten wir uns sehr herzlich bei Herrn Sven Jacobitz, Herrn Prof. Thorsten Pawletta, Frau Christina Deactu, Herrn Siddhartha Gupta sowie bei allen weiteren Beteiligten bedanken. Besonderer Dank gilt auch Herrn Prof. Oliver Rose und seinem Team für die technische Unterstützung.

Für Form und Inhalt der im vorliegenden Tagungsband enthaltenen Beiträge sind die Autoren selbst verantwortlich.

Prof. Dr.-Ing. Xiaobo Liu-Henke, apl. Prof. Dr.-Ing. habil. Umut Durak
Wolfenbüttel, im März 2021

Automatisierte Umgebungsmodellierung zur Erprobung intelligenter Fahrzeugfunktionen mittels DSIL-Simulationen

Marian Göllner^{1*}, Sven Jacobitz¹, Sören Scherler¹, Jie Zhang¹, Or Aviv Yarom¹, Xiaobo

Liu-Henke¹

¹Institut für Mechatronik, Ostfalia Hochschule für angewandte Wissenschaften, Salzdahlumer Str. 46/48, 38302 Wolfenbüttel; *mar.goellner@ostfalia.de

Abstract. Um den fehlenden Schritt zwischen Hardware-in-the-Loop-Simulation und Fahrversuch zu schließen wurde an der Ostfalia ein Konzept entwickelt, welches einen Closed-Loop-Fahrsimulator Driving-Simulator-in-the-Loop-(DSiL)-Simulation nutzt. Um eine perfekte Immersion des Testfahrers in der Fahrsimulation zu erreichen ist eine möglichst realistische Simulation der Fahrumgebung nötig. Diese wird aktuell weitgehend durch den Einsatz von Computerspielframeworks wie Unity oder Unreal anhand manueller modellierter realer Strecken realisiert. Im vorliegenden Beitrag wird eine Methode aufgezeigt, mit der zum einen die Streckenerstellung in großen Teilen automatisiert und zum anderen durch einen vereinfachten Prozess jeder beliebige Abschnitt einer realen Strecke oder Strecke digitalisiert werden kann.

Einleitung und Motivation

Der bestehende Entwurfs- und Validierungsprozess für fahrzeugmechatronische Systeme berücksichtigt erst spät den Einfluss realer Fahrer. Das reale kognitive Verhalten dieser spielt jedoch insbesondere im Zusammenhang mit Fahrerassistenzsystemen und hoch automatisiertem Fahren eine wichtige Rolle. Um diesen Einfluss frühzeitig vor dem Bau erster Prototypen hinreichend zu berücksichtigen, wird im Rahmen des Niedersächsischen Zukunftslabors Mobilität die bewährte modellbasierte mechatronische Entwurfsmethode um den Einsatz eines Closed-Loop-Fahrsimulators erweitert. Neben den Model-in-the-Loop- (MiL-), Software-in-the-Loop- (SiL-) und Hardware-in-the-Loop- (HiL-)Simulationen erfolgen

zusätzlich frühzeitig Driving-Simulator-in-the-Loop- (DSiL-)Simulationen (siehe Abbildung 1).

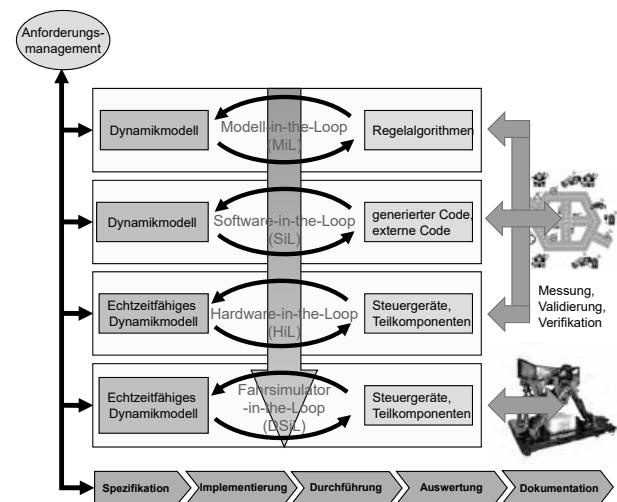


Abbildung 1: Um die DSiL-Simulation erweiterte Methode

So können, wie bereits in [1] nachgewiesen, Testszenarien realitätsnah und reproduzierbar in einer Closed-Loop-Simulation unter Einbeziehung des menschlichen Verhaltens durchgeführt werden, bevor kostenintensive und gefährliche Versuche in realen Prototypen erfolgen. Es kann bereits in einem frühen Entwicklungsstadium untersucht werden, wie ein Nutzer physiologisch und psychologisch im Zusammenspiel mit den Funktionen reagiert und wie diese durch den menschlichen Fahrer beeinflusst werden. Damit dieser Ansatz funktioniert, ist es wichtig, dass der Nutzer sich in die Simulation hineinversetzt fühlt (Immersion) und idealerweise keinen Unterschied zu einem realen Erlebnis bemerkt.

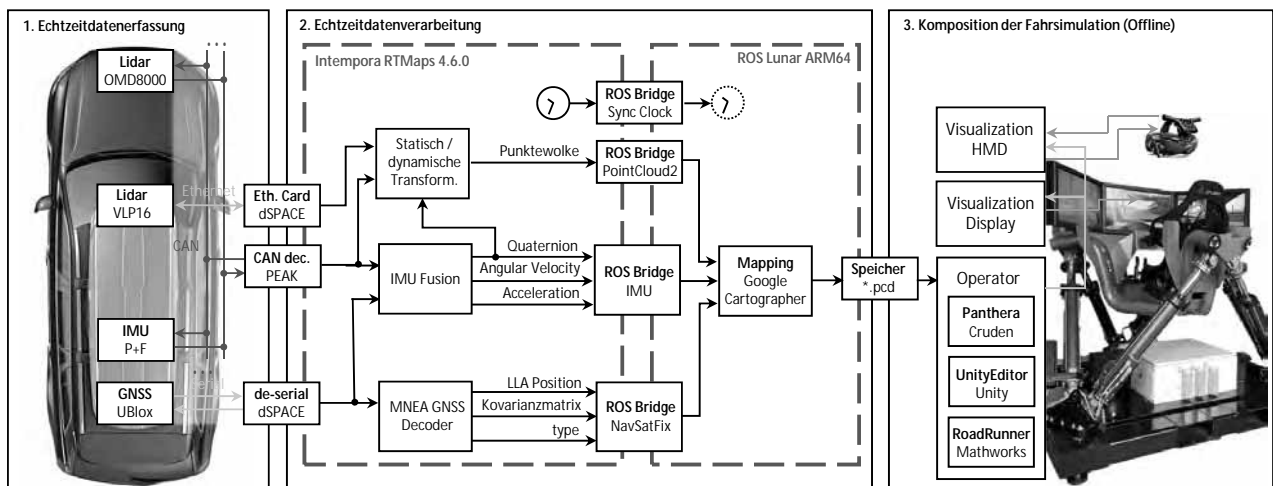


Abbildung 2: Grundlegender Ablauf der automatisierten Umfeldmodellierung

1 Konzept

Ein wichtiger Teil der Immersion ist die realitätsnahe Darstellung der Umgebung, was z. B. mittels Computerspiel-Frameworks erfolgen kann. Diese sind durch den kommerziellen Antrieb der Industrie mittlerweile entsprechen realistisch; eine Umgebung zu modellieren ist jedoch zeitaufwändig und erfordert tiefgehendes Wissen und Erfahrung. Entsprechend werden meist nur prominente Strecken digitalisiert. Doch gerade eine individuell angepasste Umgebung in welcher der Testproband sich auskennt, z.B. der tägliche Weg zu Arbeit, versprechen sowohl eine Validierung der Immersion des Simulators als auch eine realistische Abbildung realer Fahrsituationen und Probandenreaktionen. Die hier vorgestellte Methode der automatisierte Umfeldmodellierung ermöglicht genau diesen Umstand zu verbessern, indem Strecken in kurzer Zeit weitgehend automatisiert digitalisiert werden können.

Die automatisierte Umfeldmodellierung erfolgt in zweiten Schritten. Der erste Teil besteht in der Datenaufnahme, -vorverarbeitung und -speicherung mittels speziell ausgerüstetem Messfahrzeug unter Echtzeitbedingungen. Der zweite Schritt ist die Aufbereitung der Daten zu einer visuell immersiven, dreidimensionalen Abbildung der Umgebung zur Implementierung auf das Visualisierungssystem des Fahrsimulators. Besondere Anforderungen bestehen in der Darstellung der Umgebungsoberflächen (Texturierung) und der Lichtquellenanordnung, um die Umgebung möglichst realistisch erscheinen zu lassen. Abbildung 2 illustriert die übergeordnete Struktur des Grundlegenden Ablaufs.

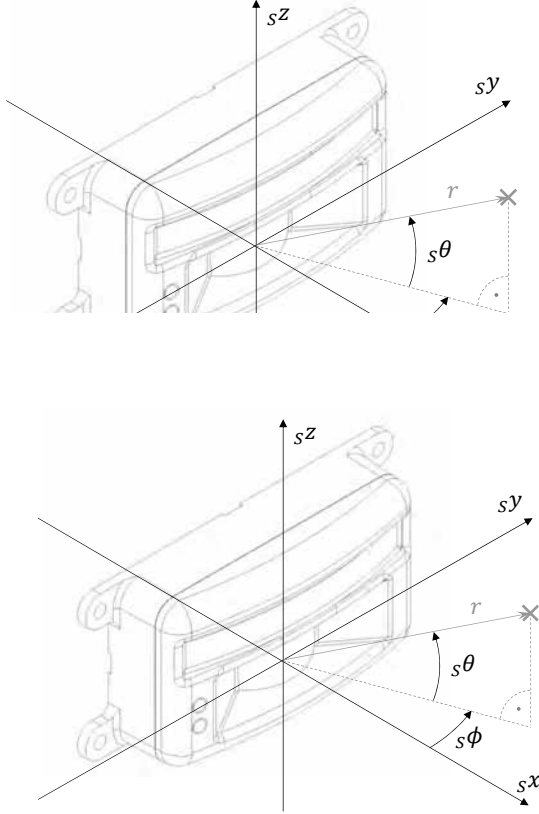
2 Echtzeitdatenerfassung

Die Echtzeitdatenerfassung erfolgt mittels eines Messfahrzeugs auf dessen Dach und an dessen Front die notwendige Sensorik angebracht wurde. Wie in Abbildung 2 zu erkennen, sind für Die Datenaufnahme mehrere Lidar- Sensoren, eine inertielle Messeinheit (IMU) und ein Satellitennavigationssystem (GNSS System) vorgesehen. Die Lidar-Sensoren tasten die Umgebung in Bezug auf ihr internen Koordinatensystem ab. Da das Fahrzeug sich selbst aber bewegt, muss in Kapitel 3.1 eine Koordinatentransformation der internen Sensorkoordinaten zunächst auf den Bezugsrahmen des Fahrzeugs und mittels IMU und inverser Kinematik auf den Bezugsrahmen der Umgebung umgerechnet werden. Die GNSS Sensorik dient sowohl zur Korrektur als auch zur Georeferenzierung dieses festen Bezugsrahmens auf das globale WGS-84 Koordinatensystem.

2.1 Lidar Sensoren

Die Erfassungssensorik besteht primär aus light detection and ranging (Lidar-) Sensoren, welche mittels Infrarotlaserstrahlen die Umgebung abtasten. Das Konzept sieht einen rotierenden fächerförmigen Laseremitter/-empfänger auf dem Dach des Messfahrzeuges und mehrere um das Fahrzeug herum gruppierte feststehende fächerförmige Laseremitter/-empfänger vor. Durch diese Anordnung ist eine maximale Abdeckung der Umgebung und somit eine hohe Detailauflösung der resultierenden Punktwolken möglich. Der rotierende Laseremitter/-empfänger ist vom Typ eines Velody-

ne VLP16, die feststehenden Laseremitter/-empfänger sind Pepperl+Fuchs OMD8000 Sensoren. Eine Grund-



2.2 Inertiale Messeinheit

Zur weiteren Verarbeitung der gewonnenen Punktwolken müssen diese in Relation zur Bewegung des Messfahrzeugs gesetzt werden. Dazu wird eine inertielle Messeinheit (IMU) verwendet, welche aus grundlegenden Beschleunigungs-, Gierraten- und Magnetfeldsensoren aufgebaut ist. Die Abgabedaten der Grundsensoren, explizit die lineare Beschleunigung in alle Raumrichtungen ${}_M\ddot{x}_t, {}_M\ddot{y}_t, {}_M\ddot{z}_t$, die Drehraten um die Hauptkoordinatenachsen des Sensorkoordinatensystems ${}_M\dot{\phi}_t, {}_M\dot{\theta}_t, {}_M\dot{\psi}_t$ und der Magnetfeldvektor \vec{g} jeweils zu einem Zeitschritt t , werden über einen Fusionsalgorithmus gefiltert, driftbereinigt und zu den für

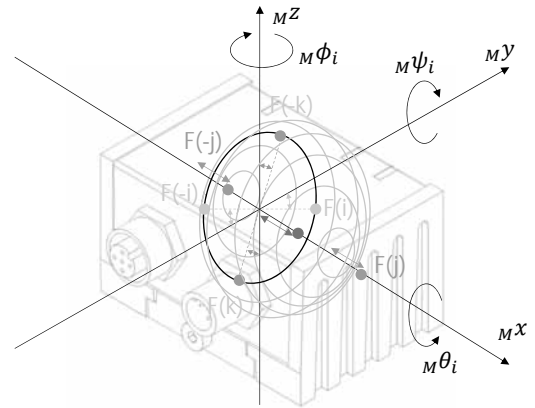


Abbildung 4: Koordinatensystem der IMU mit Drehwinkeln und der in den dreidimensionalen Raum projizierten Hypersphäre des Quaternion ${}_Mq$

als Ursprung, umgerechnet werden können.

$$\begin{bmatrix} {}_sx_i \\ {}_sy_i \\ {}_sz_i \end{bmatrix} = \begin{bmatrix} \sin {}_s\theta \cdot \cos {}_s\phi \\ \sin {}_s\theta \cdot \sin {}_s\phi \\ \cos {}_s\theta \end{bmatrix} \cdot r \quad (1)$$

Die Emmitter des Velodyne VLP-16 Sensor sind in festen Winkeln zu ${}_s\theta$ angeordnet und rotieren kontinuierlich um den Winkel ${}_s\phi$. Die Emmitter des Pepperl+Fuchs OMD8000 sind hingegen in festen Winkeln zu ${}_s\phi$ angeordnet, rotieren nicht und sind alle in selbem Winkel ${}_s\theta = 0$ aufgereiht. Die ausgegebenen Punktwolken sind auf das jeweilige Sensorkoordinatensystem referenziert und durch interne Timer mit einem Zeitstempel versehen; Sie werden in fester Taktrate übertragen.

Auf den Fusionsalgorithmus soll hier nur kurz eingegangen werden. Für weiter Informationen siehe [3]. Die Struktur des verwendeten Kalman-Filter basiert auf der durch den Gierratensensor gemessenen Winkeländerung in einem durch die Abtastrate f_s festgelegten Zeitschritt:

$$\begin{bmatrix} \Delta {}_M\phi_i \\ \Delta {}_M\theta_i \\ \Delta {}_M\psi_i \end{bmatrix} = \begin{bmatrix} {}_M\dot{\phi}_t - {}_M\dot{\phi}_{t-1} \\ {}_M\dot{\theta}_t - {}_M\dot{\theta}_{t-1} \\ {}_M\dot{\psi}_t - {}_M\dot{\psi}_{t-1} \end{bmatrix} \cdot f_s^{-1} \quad (2)$$

Diese werden mittels der gemessenen Beschleunigungen auf eine Ruhelagenreferenz in Richtung Erdmagnetfeldvektor korrigiert, der Winkel um die senkrechte

Achse ist entsprechend nicht korrigierbar:

$$\begin{bmatrix} \Delta_M \phi \\ \Delta_M \theta \\ \Delta_M \psi \end{bmatrix} = \begin{bmatrix} \Delta_M \phi_i \\ \Delta_M \theta_i \\ \Delta_M \psi_i \end{bmatrix} + \begin{bmatrix} \arctan \frac{M y_i}{\sqrt{M x_i^2 + M z_i^2}} \cdot \frac{180}{\pi} \\ \arctan \frac{M x_i}{\sqrt{M y_i^2 + M z_i^2}} \cdot \frac{180}{\pi} \\ 0 \end{bmatrix} \quad (3)$$

Die errechneten Winkel werden anschließend in eine Quaternionendarstellung $Mq = q_1 + iq_2 + jq_3 + kq_4$ mit den Parametern $i^2 = j^2 = k^2 = i \cdot j \cdot k = -1$ überführt. Dies hat den Vorteil, dass sowohl die Effizienz der Rotationsoperation steigt als auch die Komplexität des zugrundeliegenden Algorithmus sinkt [4]. Die Abbildung $\rho_q : x \mapsto qxq^{-1} = qxq^*$ auf $\text{Im } \mathbb{H}$ beschreibt im dreidimensionalen Raum eine Drehung. Aufgrund dieses einfachen linearen Zusammenhangs ist eine lineare Interpolation (Slerp: spherical linear interpolation) möglich, was wiederum die Vorhersagegenauigkeit des Kalman Filter verbessert. Der Drehvorgang ist vierdimensional, eine projizierte Darstellung der Drehung ist in Abbildung 4 angedeutet. Die Drehung im Sensorbezugsrahmen lässt sich mit Gleichung bestimmen:

$$M\Delta q_i = \begin{bmatrix} \cos \frac{M\Delta\psi}{2} \\ 0 \\ 0 \\ \sin \frac{M\Delta\psi}{2} \end{bmatrix} \begin{bmatrix} \cos \frac{M\Delta\theta}{2} \\ 0 \\ \sin \frac{M\Delta\theta}{2} \\ 0 \end{bmatrix}$$

Die absolute Orientierung wird durchschritt auf Basis der in vorherigen Schritt ein Modell präzidierten Orientierung

$$Mq_t = Mq_{t-1} \prod_{i=1}^n M^L$$

Der Gravitationsvektor kann nach dem Quaternion betrachtet werden. Dieser wird durch Subtraktion der abgeklärten Beschleunigungen der vorherigen Iteration geschätzt, da die Gravitation im vorliegenden Fall einer Fahrt auf ebener Fahrbahn durch den hohen Absolutwert eindeutig unterscheidbar von den horizontalen Beschleunigungen ist. Der geschätzte Vektor wird abschließend durch die Messwerte des Magnetometers verbessert, sodass abschließend eine zuverlässige Aussage über die Orientierung in Bezug zum Erdmittelpunkt getroffen werden kann.

2.3 GNSS Messeinheit

Um die Punktwolke global zu referenzieren und so das Zusammenführen mehrerer Datenerhebungen, also Messfahrten, zu ermöglichen, wird eine GNSS Messeinheit in das System integriert. Die Ausgabewerte dieser werden in das lokale Koordinatensystem umgerechnet und als Offset-Parameter den Punktwolken hinzugefügt. Die Positionsmessung im globalen Referenzsystem des WGS-84 Ellipsoiden basiert auf der Signallaufzeit $\Delta t_{Mi} = \Delta t_i + \Delta t_{i0}$ der GNSS Satelliten $i = 1 \dots n$ deren Position im Erdborbit $X_{S_i}, Y_{S_i}, Z_{S_i}$ zum Zeitpunkt des Signalempfangs bekannt sind. Mindestens 4 Satelliten müssen zur Positionsbestimmung gleichzeitig sichtbar sein. Aus den Signallaufzeiten zum Empfänger lassen sich mittels der Signalgeschwindigkeit (Lichtgeschwindigkeit im Vakuum c) Radien $R_1 \dots R_n$ einzelner Empfangssphären berechnen, bei deren gemeinsamem

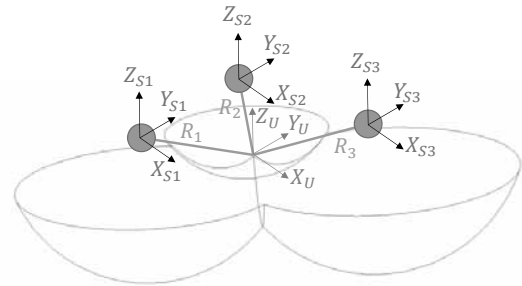


Abbildung 5: Koordinatensysteme der Satelliten $X_{S_i}, Y_{S_i}, Z_{S_i}$ und des Empfängers X_U, Y_U, Z_U | Schnittmenge der Empfangssphären mit den Radien $R_1 \dots R_n$

Die entsprechenden Radien der Empfangssphären lassen sich aus den Positionen der Satelliten und der noch zu ermittelnden Position des Nutzers berechnen.

$$R_i = \sqrt{(X_{S_i} - X_U)^2 + (Y_{S_i} - Y_U)^2 + (Z_{S_i} - Z_U)^2}$$

Da die Signallaufzeiten als Differenz der Zeitgeber der Satelliten zu dem Zeitgeber des Empfängers kalkuliert werden, ist eine Zeitdifferenz aufgrund von Ungenauigkeiten und Fehlern in jedem Falle zu berücksichtigen, sodass eine Pseudorange R_p als Näherung der exakten

Position angenommen wird.

$$R_{P_i} = R_i + \Delta t_{i0} \cdot c \quad (6)$$

Unter der Annahme das 4 Satelliten sichtbar sind und durch eine Normierung auf $R_i = \Delta t_{Mi} \cdot c$ erhält man die Positionsmatrix \underline{A} :

$$\begin{bmatrix} R_{P_1} - R_1 \\ R_{P_2} - R_2 \\ R_{P_3} - R_3 \\ R_{P_4} - R_4 \end{bmatrix} = \underbrace{\begin{bmatrix} \frac{X_{S_1} - X_U}{R_1} & \frac{Y_{S_1} - Y_U}{R_1} & \frac{Z_{S_1} - Z_U}{R_1} & -1 \\ \frac{X_{S_2} - X_U}{R_2} & \frac{Y_{S_2} - Y_U}{R_2} & \frac{Z_{S_2} - Z_U}{R_2} & -1 \\ \frac{X_{S_3} - X_U}{R_3} & \frac{Y_{S_3} - Y_U}{R_3} & \frac{Z_{S_3} - Z_U}{R_3} & -1 \\ \frac{X_{S_4} - X_U}{R_4} & \frac{Y_{S_4} - Y_U}{R_4} & \frac{Z_{S_4} - Z_U}{R_4} & -1 \end{bmatrix}}_{\underline{A}} \cdot \begin{bmatrix} \Delta x \\ \Delta y \\ \Delta z \\ \Delta t_0 \end{bmatrix}$$

Die exakte Position X_{eU}, Y_{eU}, Z_{eU} im ECEF erhält man durch Korrektur um die Fehler $\Delta x, \Delta y, \Delta z, \Delta t_0$:

$$\begin{aligned} X_{eU} &= X_U + \Delta x \\ Y_{eU} &= Y_U + \Delta y \\ Z_{eU} &= Z_U + \Delta z \end{aligned} \quad (7)$$

Typischerweise werden die Fehler in der aus \underline{A} abgeleiteten Kovarianzmatrix von einem kalibrierten Empfänger übergeben [5]:

$$\underline{Q} = (\underline{A}^T \underline{A})^{-1} = \begin{bmatrix} \sigma_x^2 & \sigma_{xy} & \sigma_{xz} & \sigma_{xt} \\ \sigma_{xy} & \sigma_y^2 & \sigma_{yz} & \sigma_{yt} \\ \sigma_{xz} & \sigma_{yz} & \sigma_z^2 & \sigma_{zt} \\ \sigma_{xt} & \sigma_{yt} & \sigma_{zt} & \sigma_t^2 \end{bmatrix} \quad (8)$$

Zumindest die Verringerung der Genauigkeit (Dilution of precision, DOP) in horizontaler ($HDOP = \sqrt{\sigma_x^2 + \sigma_y^2}$) und vertikaler ($VDOP = \sqrt{\sigma_z^2}$) Richtung werden durch die NMEA Norm 0183 als Ausgabestandardnachricht vorgegeben. Aus den gewonnenen Koordinaten im ECEF können nun Längengrad λ und Breitengrad φ transformiert werden. Der Längengrad λ ist durch die folgende einfache Beziehung bestimmt [6]:

$$\lambda = \arctan \frac{Y_{eU}}{X_{eU}} \quad (9)$$

Der Breitengrad ist nur durch iterative Berechnung zu erhalten. Dafür spielen die numerischen Exzentrizitäten ε und $\tilde{\varepsilon}$ der Halbachsen des Weltellipsoiden (WGS-84) $a = 6.378.137,00m$ und $b = 6.356.752,31m$ eine Rolle

$$\varepsilon = \frac{\sqrt{a^2 - b^2}}{a}, \tilde{\varepsilon} = \frac{\sqrt{a^2 - b^2}}{b} \quad (10)$$

Es gilt unter Einbezug der parametrischen Breite t mit dem Zusammenhang $\tan \varphi = \sqrt{1 - \varepsilon^2} \cdot \tan t$ ungefähr:

$$\tan \varphi \approx \frac{Z_{eU} + \tilde{\varepsilon}^2 b \sin^3 t}{X_{eU} + \varepsilon^2 a \cos^3 t} \quad (11)$$

Die Höhe über dem Referenz- Ellipsoid h ist über den Krümmungsradius

$$N_\varphi = \frac{a}{\sqrt{1 - \varepsilon^2 \sin^2 \varphi}} \quad (12)$$

aus den gefunden Längen und Breitengrad abzuleiten:

$$h = \frac{\sqrt{X_{eU}^2 + Y_{eU}^2}}{\cos \varphi} - N_\varphi \quad (13)$$

Die elliptischen Koordinaten Länge φ , Breite λ und Höhe über dem Referenz- Ellipsoid h werden über folgenden Zusammenhang auf die kartesische Positionen G^x, G^y, G^z umgerechnet [7]:

$$\begin{aligned} G^x &= (N_\varphi + h) \cdot \cos \varphi \cdot \cos \lambda \\ G^y &= (N_\varphi + h) \cdot \cos \varphi \cdot \sin \lambda \\ G^z &= (N_\varphi \cdot (1 - \varepsilon^2) + h) \cdot \cos \varphi \end{aligned} \quad (14)$$

3 Echtzeitdatenverarbeitung

Ziel ist zunächst in jedem Abtastschritt eine fusionierte Punktwolke mit global referenzierten Punkten zu erhalten. Die kontinuierliche Fahrt durch die Umgebung mit festen Abtastschritten erzeugt dann einen Satz solcher Punktwolken, welche mittels Mapping und Optimierung derer Transformationsmatrizen zu einer größeren Punktwolke fusioniert wird. Diese wird nach Abfahren der festgelegten Route in einem allgemeinen Austauschformat als *.pcd Datei gespeichert und dem Offline Datenverarbeitungsschritt übergeben.

3.1 Datentransformation

Jedes der sensorischen Datenerfassungssystem arbeitet zunächst mit dem eigenen, internen Koordinatensystem; Die Sensorausgänge tragen eigene interne Zeitstempel. Zunächst werden durch die Nutzung des Frameworks RTMaps [8] alle Sensordaten gesammelt und über deren Zeitstempel synchronisiert. Ein neuer Zeitstempel wird auf Basis der GNSS Daten generiert.

Die Lage der einzelnen Sensoren muss für die nachfolgenden Datenfusion statisch auf das Fahrzeugkoordinatensystem (BCS) und dynamisch auf ein feststehendes

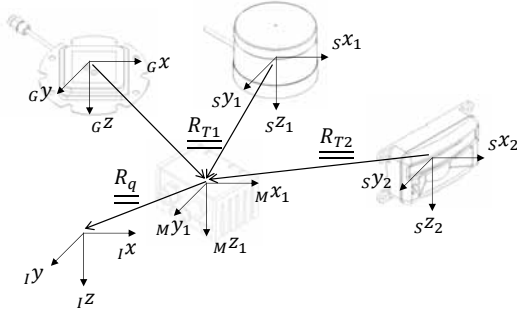


Abbildung 6: Koordinatentransformationen

Die statische Transformation der Koordinatensysteme um fixe rotatorische Verdrehungen zu dem Koordinatensystem des Fahrzeugs $^FXS, ^FYs, ^FzS$ (BCS) mit den Achsenwinkeln $^F\phi, ^F\theta, ^F\psi$ unter der Bedingung gleicher SI Einheiten ist über die folgende Gleichung möglich.

$$\begin{aligned} \underline{\underline{R_x}}(^F\psi) &= \begin{bmatrix} \cos_F\psi & -\sin_F\psi & 0 \\ \sin_F\psi & \cos_F\psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \\ \underline{\underline{R_y}}(^F\theta) &= \begin{bmatrix} \cos_F\theta & 0 & \sin_F\theta \\ 0 & 1 & 0 \\ -\sin_F\theta & 0 & \cos_F\theta \end{bmatrix} \\ \underline{\underline{R_z}}(^F\phi) &= \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos_F\phi & -\sin_F\phi \\ 0 & \sin_F\phi & \cos_F\phi \end{bmatrix} \end{aligned} \quad (15)$$

Es kann eine Rotationsmatrix R_{rot} abgeleitet werden:

$$R_{rot}(^F\phi, ^F\theta, ^F\psi) = R_z(^F\phi) \cdot R_y(^F\theta) \cdot R_x(^F\psi) \quad (16)$$

Die statische Translation ist über den Vektor R_{tra} möglich:

$$R_{tra}(^FX, ^FY, ^Fz) = \begin{bmatrix} ^FX & ^FY & ^Fz \end{bmatrix}^T \quad (17)$$

So kann eine Matrize für die statisch rotatorische und translatorische Transformation hergeleitet werden. Die Umrechnung soll in diesem Falle über die leicht zu-

gänglichen Euler-Winkel geschehen, da dieser Schritt nur ein einziges Mal zur Kalibrierung der zueinander feststehenden Koordinatensysteme der Sensoren und des BCS nötig ist. Die Transformation der Lidar-Punktwolke wird entsprechend über die folgende Gleichung durchgeführt:

$$\begin{bmatrix} ^FXS \\ ^FYs \\ ^FzS \end{bmatrix} = \underbrace{\begin{bmatrix} R_{rot}(^F\phi, ^F\theta, ^F\psi) & R_{tra}(^FX, ^FY, ^Fz) \end{bmatrix}}_{\underline{\underline{R_{Ti}}}} \cdot \begin{bmatrix} sX \\ sY \\ sZ \\ 1 \end{bmatrix}$$

Da das Messfahrzeug sich fortwährend bewegt, sollen die Punktwolken auch dynamisch transformiert werden. Die dynamische Transformation der Rotation soll kontinuierlich erfolgen, weshalb sie auf Basis des von der IMU ausgegeben Quaternion durchgeführt wird. Der Einfachheit halber wird deshalb das Koordinatensystem des Fahrzeugs $^FXS, ^FYs, ^FzS$ (BCS) mit dem Koordinatensystem der IMU $^MX, ^MY, ^MZ$ gleichgesetzt. Die Transformation erfolgt auf das feststehende Inertialkoordinatensystem $^IX, ^IY, ^IZ$ (ICS). Die Beschreibung einer allgemeinen Rotation ist entsprechend über die folgenden Transformationsmatrizen möglich:

$$\begin{aligned} \underline{\underline{R_{zp}}}(mq) &= \begin{bmatrix} 2 \cdot (q_2 \cdot q_4 + q_1 \cdot q_3) \\ 2 \cdot (q_3 \cdot q_4 - q_1 \cdot q_2) \\ 1 - 2 \cdot (q_2^2 + q_3^2) \end{bmatrix} \\ \underline{\underline{R_{yp}}}(mq) &= \begin{bmatrix} 2 \cdot (q_2 \cdot q_3 - q_1 \cdot q_4) \\ 1 - 2 \cdot (q_4^2 + q_2^2) \\ 2 \cdot (q_3 \cdot q_4 + q_1 \cdot q_2) \end{bmatrix} \\ \underline{\underline{R_{xp}}}(mq) &= \begin{bmatrix} 1 - 2 \cdot (q_3^2 + q_4^2) \\ 2 \cdot (q_2 \cdot q_3 + q_1 \cdot q_4) \\ 2 \cdot (q_2 \cdot q_4 - q_1 \cdot q_3) \end{bmatrix} \end{aligned} \quad (18)$$

Die dynamische Transformation der im vorherigen Schritt gewandelten Sensorpunkte erfolgt so über die Quaternion der IMU über:

$$\begin{bmatrix} ^IX \\ ^IY \\ ^IZ \end{bmatrix} = \underbrace{\begin{bmatrix} R_{xp}(mq) & R_{yp}(mq) & R_{zp}(mq) \end{bmatrix}}_{\underline{\underline{R_q}}} \cdot \begin{bmatrix} ^FXS \\ ^FYs \\ ^FzS \end{bmatrix}$$

In die dynamische Transformation der Translation werden zudem die GNSS-Daten integriert. Das Prinzip ist dem der vorgestellten statischen Translation gleich. Es werden sowohl die fusionierte Punktwolke (Point-Cloud2) als auch die Quaternion, lineare Beschleunigung und Winkelgeschwindigkeit als Zustände aus der

IMU (IMU) und auch die globale Position in Längengrad, Breitengrad und Höhe über dem Referenz- Ellipsoid mit der zugehörigen Kovarianz- Matrix (NavSat-Fix) an den nächsten Schritt als ROS Messages übergeben. Weiterhin wird die ROS Clock mit der RTMaps Clock synchronisiert, wodurch alle generierten Zeitstempel Genauigkeit und Gültigkeit behalten.

3.2 Fusion der Punktwolken

Werden zwei zu jeweils nachfolgenden Zeitschritten aufgenommene Punktwolken mit selber Referenzierung übereinander gelegt, so können Punkte, die dieselbe Geometrie im Raum abtasten unter Berücksichtigung eines Offsets, der durch lineare und rotatorische Bewegung hervorgerufen wird, als gleiche Punkte identifiziert werden. Schwierigkeit ist die optimale Rotation und Translation der aufeinanderfolgenden Punktwolken j und k unter der Annahme der Detektionspunkte als nearest Neighbors in Form eines Tuple (m_i, d_i) zu finden. Zu diesem Zwecke kann eine Optimierungsfunktion aufgestellt werden, die den jeweils optimalen Rotationsvektor einer Punktwolke $R_{T_{i_{opt}}} = (R_{T_i}, t)$ zurückgibt [9]. Die Optimierung wird über einen Fehlervektor E , welcher von $R_{T_{i_{opt}}}$ abhängt, durchgeführt:

$$E(R_{T_i}, t) = \frac{1}{n} \sum_{i=1}^n \|m_i - (R_{T_i} \cdot d_i + t)\|^2 \quad (19)$$

Für zwei aufeinanderfolgende Punktwolken ergibt sich so:

$$E = \sum_{j \rightarrow k} \sum_i \|R_{T_j} \cdot m_i + t_j - (R_{T_k} \cdot d_i + t_k)\|^2 \quad (20)$$

Durch eine Minimierung dieser Fehlerfunktion über alle aufgenommenen Punktwolken unter Berücksichtigung der zu dem jeweiligen Zeitschritt zugeordneten Bewegungsreferenzen ergibt sich eine gesamte Abbildung der Messfahrt als fusionierte Punktwolke, welche abschließend als *.pcd Datei abgelegt wird. Dieser Schritt wird während der Datenerhebung kontinuierlich mit Hilfe des google cartographer [10] in ROS ausgeführt. Der in dem Austauschformat gespeicherte Datensatz ist eine allgemeine, dreidimensionale Beschreibung der Umgebung, die nun für die Darstellung im Fahrsimulator aufbreitet werden muss. Dazu werden die Daten in der vom Visualisierungssystem Panthera verwendeten Unity Engine editiert.

4 Komposition der Simulation

4.1 Erstellen des Umgebungsmodells

Die einzelnen Punkte der PointCloud werden als Eckpunkte (Vertices) von Grundobjekten (Grafischen Primitiven) wie Polygonen, in einem georeferenzierten homogenen Koordinatensystem definiert. Vertex- Elemente, die in einem definierten engen Raster zueinander stehen, werden automatisch mit Primitiven verbunden, so das ein Netz entsteht, welches wie ein Drahtgitter (Mesh) die Oberflächenkontur der Umgebung wiedergibt [11]. Auf diesem Drahtgitter werden Texturen aufgelegt, welche der Optik des Umgebungsmaterials entsprechen. Durch das Mapping werden Vertex, Mesh und Textur verbunden und zu einer dreidimensionalen Fläche, welche nun durch Shading mit Licht-, Schatten- und Tiefeneffekte versehen werden kann. Dieser Schritt wird mit Hilfe des RoadRunner Streckeneditors realisiert, welcher die notwendigen Schnittstellen und Berechnungsmethoden aufweist um die vorgestellten Arbeitsschritte ausführen zu können. Die dreidimensionale Umgebung wird abschließend zu Unity exportiert. Die letztendliche Echtzeitdarstellung des Bildes während der Simulation im Fahrsimulator wird ausgehend von der Kameraposition gerendert und ist ein hier nachgelagerter Prozess.

4.2 Integration in das Visualisierungssystem

In diesem Schritt werden zunächst Koordinaten für das initiale Platzieren (Spawnen) von dynamischen Objekte wie z.B. dem Egofahrzeug oder anderen Verkehrsteilnehmern als auch Kamerapositionen des Fahrers oder zur Beobachtung des Geschehens festgelegt. Dies geschieht durch einbetten von einfachen Referenzobjekten, deren Label von dem Visualisierungssystem Panthera durch abgespeicherte Assets ersetzt werden.

Weiterhin werden Oberflächeneigenschaften anhand der begrenzenden Primitive auf die Oberflächenkontur definiert. So kann das Visualisierungssystem virtuelle Fahrbahnen (Straßen, Bürgersteige, etc.) mit passenden realistischen Unebenheiten und Rauheiten belegen, sodass die Abrolleigenschaften und der physikalische Rad-Straße-Kontakt wie in [12] beschrieben berechnet werden können. Die vorgestellten Arbeitsschritte werden direkt in der Unity Engine mithilfe eines Editors durchgeführt. Das Ergebnis wird nun in das Visualisierungssystem integriert und auf allen beteiligten Rechnersystemen synchronisiert [13].

5 Test und Validierung

Im Folgenden werden Teilergebnisse der vorgestellten automatisierten Umfeldmodellierung dargestellt. Nachfolgend zeigt die Abbildung 7 zunächst die Rohdaten die von dem Echtzeitprozess an den nachgelagerten Modellierungsschritt übergeben werden.

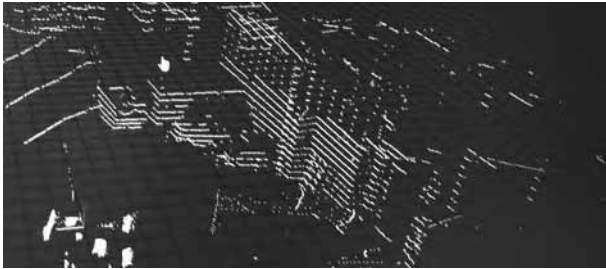


Abbildung 7: Ergebniss Echtzeitdatenverarbeitung

Der Modellierungsschritt ist zum Vergleich in der folgenden Abbildung 8 dargestellt. Die Texturierung wird noch händisch erstellt. Orange Markiert die extrahierten Straßenzüge.

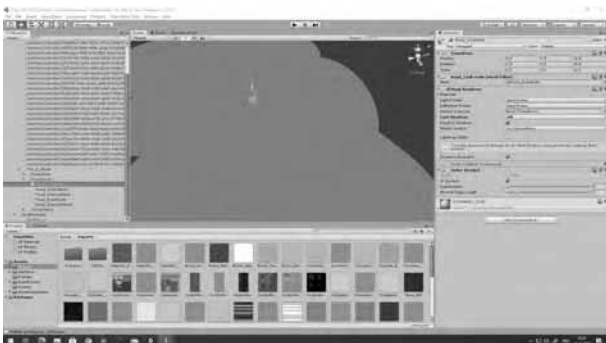


Abbildung 8: modelliertes Umgebungsmodell

Abschließend zeigt Abbildung 9 die fertig modellierte Umgebung durch das Visualisierungssystem des Fahrsimulators gerendert.



Abbildung 9: Umgebung in Fahrsimulation

Zusammenfassend ist jeder Teilschritt funktional und zumindest so genau, dass ein Testproband einen hohen Wiedererkennungseffekt hat und seine Umgebung als Vertraut wahrnimmt. Folgende Validierungsschritte werden die Durchgängigkeit und die allgemeine Adaptierbarkeit des Ansatzes prüfen. Diese waren zum Zeitpunkt der Erstellung dieser Schrift noch nicht abgeschlossen.

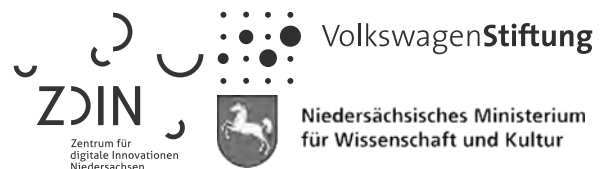
6 Zusammenfassung

Im vorliegende Beitrag wurden sowohl das Konzept als auch das realisierte System einer automatisierten Umfeldmodellierung zur Erprobung intelligenter Fahrzeugfunktionen mittels DSIL-Simulationen vorgestellt. Es wurde aufgezeigt wie ein reales (Fahr-)Umfeld mittels Sensorik aufgenommen und gespeichert werden kann. Wie aus diesen gespeicherten Daten, in diesem Falle Punktwolken, eine digitalisierte Umgebung mittels Technologien der Computerspieleindustrie mit wenig händischen Aufwand erstellt und abschließend so referenziert werden kann, dass das Zielsystem, der Closed-Loop Fahr Simulator, das Umfeld als 3D Graphik rendern und dynamisch darstellen kann. Zur Validierung des Systems wurde ein Messfahrzeug aufgebaut und mit der notwendigen Sensorik ausgerüstet. Zudem wurde eine Toolkette implementiert mit der die Datenaufnahme und Vorverarbeitung Online auf dem Fahrzeug und Offline auf dem Operator PC des Fahr Simulators durchgeführt wird.

Weiterhin ist der Schritt der Texturierung noch nicht automatisiert, da eine sensorische Quelle fehlt. Zukünftig soll ein Kamerasystem integriert werden, welches diese Daten liefert.

Danksagung

Gefördert vom Niedersächsischen Ministerium für Wissenschaft und Kultur unter Fördernummer ZN3495 im Niedersächsischen Vorab der VolkswagenStiftung und betreut vom Zentrum für digitale Innovationen (ZDIN).



Referenzen

- [1] Göllner M, Tao H. Simulator development. *Automotive Testing Technology International*. 2018; (September):102–106.
URL <https://www.ukimediaevents.com/publication/b913862a/104>
- [2] Kannan K. Development of a reference software platform for the Velodyne VLP-16 LiDARS. Master's thesis, KTH Royal Institute of Technology, Stockholm. 2016.
- [3] Solà J. Quaternion kinematics for the error-state Kalman filter.
URL <https://arxiv.org/pdf/1711.02508>
- [4] Shoemaker K. Animating Rotation with Quaternion Curves. In: *SIGGRAPH 85 conference proceedings, July 22-26, San Francisco*, edited by Barsky BA, Computer Graphics, pp. 245–254. Baltimore: Association for Computer Machinery. 1985;.
- [5] Axell E, Johansson P, Alexandersson M, Rantakokko J. Estimation of the position error in GPS receivers.
- [6] Osen K. Accurate Conversion of Earth-Fixed Earth-Centered Coordinates to Geodetic Coordinates. Ph.D. thesis, Norwegian University of Science and Technology. 2017.
URL <https://hal.archives-ouvertes.fr/hal-01704943v2>
- [7] Zogg JM. GPS: Essentials of Satellite Navigation.
- [8] Abuhadrous I, Nashashibi F, Laugeau C, Chinchole M. Multi-sensor data fusion for land vehicle localization using /sup RT/MAPS. In: *IEEE IV2003 Intelligent Vehicles Symposium*. [Piscataway, N.J.]: IEEE. 2002; pp. 339–344.
- [9] Nüchter A, Bleier M, Schauer J, Janotta P. IMPROVING GOOGLE'S CARTOGRAPHER 3D MAPPING BY CONTINUOUS-TIME SLAM. *ISPRS - International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*. 2017; XLII-2/W3:543–549.
- [10] Hess W, Kohler D, Rapp H, Andor D. Real-time loop closure in 2D LIDAR SLAM. In: *2016 IEEE International Conference on Robotics and Automation (ICRA)*. [Place of publication not identified]: IEEE. 2016; pp. 1271–1278.
- [11] Obert N. 3d Road Surface Meshing with LIDAR. Masterthesis, George Mason University, Fairfax, VA. 2015.
- [12] Liu-Henke X, Göllner M, Buchta R, Quantmeyer F, Tao H. Systemkonzept eines modularen HiL-Systems für modellbasierte Funktionsentwicklung fahr-zeugmechatronischer Systeme. In: *ASIM-Treffen STS/GMMS 2017*, edited by Commerell W, Pawletta T, ASIM Mitteilung, pp. 199–204. Wien: ARGESIM Verlag. 2017;.
- [13] Liu-Henke X, Göllner M, Jacobitz S, Scherler S, Zhang J, Yarom O. Driving-Simulator-in-the-Loop - Virtual Function Design with Consideration of Human Behaviour. In: *2020 IEEE International Symposium on Systems Engineering (ISSE) (IEEE ISSE 2020)*. Vienna, Austria. 2020; .

Presentation of a Model Integration Concept for Real Time Simulation

Oliver Ellis^{1*}

¹Institute of Flight Systems, German Aerospace Center (DLR), Lilienthalplatz 7,
38108 Braunschweig, Germany; *oliver.ellis@dlr.de

Abstract

Developing and integrating a model for a simulation system is a challenge. This process can be broken down to a consecutive progression of steps. During a modelling step, the model designer sets up a project and creates the model, thereby using a designated tool. In the next step, the model must be transformed into a format that is readable for a targeted simulator so that it can be executed. Thereby it may be necessary that the compiled model is compatible with provided dependencies that possibly be needed to be exported. Between these two steps, may include intermediate steps.

This underlying schema can also be identified at the work with the Air Vehicle Simulator (AVES) at the German Aerospace Center (DLR) in Braunschweig. AVES can run models on two targets. Both the execution with Windows and with the Unix-like real-time operating system QNX is possible. To deploy a model in AVES, it is first designed with MATLAB / Simulink. With MATLAB Coder C++ source code is generated from the Simulink model. At this stage, the generated source code is now linked against the exported dependencies that are provided by the AVES operation team and then build to a target application. This target application can now be used in the simulator.

This model integration process is subject to efforts for standardisation in order to keep it as uniform as possible. A first attempt involves a prepared MATLAB-script as a template that automates the process from the Simulink model to the target application as much as possible. The modellers can copy the file and edit it by setting configurations like a project name, step time and IP addresses. However, two problems are revealed here. Model designers may be inexperienced in software development and prefer a solution that enhances the user experience in the model integration process. Therefore, extended support might be necessary. For both the designer and the AVES team, this can disrupt the work-

flow. The other problem is that designers may change the script so that the support is then complicated. This set of individual scripts decreases the uniformity.

To address the above discussed problems the development of the AVES Model Builder is initiated. The AVES Model Builder provides a GUI over which a model designer handles the model integration process. Over the GUI the designer is able to create a profile at first. A profile is a JSON-file that contains individual information of the designed model which include the location of the Simulink model, the project name and a list of dependencies. The profile is then stored in the folder of the Simulink project file. So the AVES Model Builder can load a model by reading the profile. During the configuration the designer can choose for which target the project should be build and in which MATLAB version the Simulink model is designed. Any further MATLAB-code can be implemented in an initial script, that is referenced in the profile. The AVES Model Builder generates at some point in the process a standard MATLAB-script that references to these additions. This way a uniform method is established. Moreover, by eliminating the necessity to directly edit a template script through establishing a GUI, the user experience should be improved.

For the integration, the AVES Model Builder provides aside from Load Profile and Configure Profile four options as buttons, export dependencies, generate code from the Simulink model, build a target application and run all. The first three options correspond to the model integration steps discussed earlier. Run all groups these three options together so that with one click all three steps are run in one step. Export dependencies downloads libraries that are linked against the generated source code. Generate code opens a MATLAB console to invoke MATLAB Coder for the source code generation. In the process the MATLAB console runs the earlier mentioned generated standard script. The build target option is responsible for building the

target application. This step is comprised of intermediate steps which may vary depending on the target. A build for Windows include the build automation software CMake that generates a Microsoft Visual Studio project. Then the AVES Model Builder invokes the Microsoft build system MSBuild to compile the source code. In case for a QNX target build, a project template is used. The AVES Model Builder copies the template to its designated location and inserts information from the profile. Then the QNX build system mkbuild is invoked. In the end, a binary file is available that can be deployed into the simulator.

The AVES Model Builder also provides a headless mode, so it can be used with arguments in the command line. This provides the possibility to invoke it for continuous integration.

At the time of submission of this work, the development is still ongoing and further features yet to be implemented. In the current state the AVES Model Builder is a local application that is started by the model designer. However, for this software to work, it requires other software to be installed locally. Because of the strict licensing policy of Mathworks for MATLAB, Simulink and MATLAB Coder, it is a problem that a designer may not have a local installation on their working computer. Therefore, for future development it is planned that the AVES Model Builder provides an option for client functionality to communicate with a designated build server that consolidate the necessary software at a central spot. The model integration process is then executed remotely. Furthermore, it is planned to implement features for model validation. Also smaller features like versioning and other comfort features are scheduled. In the longer term, the AVES Model Builder is intended to become platform agnostic so that it is usable for other simulation systems than AVES.

In conclusion this work has identified the potential to enhance the user experience for the model integration process. Uniformity is created by by creating a profile that contains the necessary information. The AVES Model Builder then generates a standard script that references to the profile information. It furthermore executes the three steps of exporting dependencies, the generating code and building a target application that can be run in a simulation. The established GUI is an improvement in comparison to editing a script. It also provides a headless mode for the usage in the command line so it is usable for continuous integration. To address the strict licensing policy of Mathworks, it is

planned to integrate a client / server solution to consolidate the needed resources. All in all, this project is still a work in progress.

Mobiles Labor auf Open Source Basis für die Ausbildung mechatronischer Antriebssysteme

Rolf Roskam¹

¹Ostfalia Hochschule für angewandte Wissenschaften, Salzdhahumer Straße 46-48, 38302 Wolfenbüttel, Deutschland; r.roskam@ostfalia.de

Abstract. Bedingt durch die Corona-Pandemie, in der die Präsenzlabor für die Ausbildung nicht, oder nur sehr eingeschränkt nutzbar sind, wird hier eine mikrocontrollerbasierte Alternative zur häufig verwendeten Werkzeugkette Matlab/Simulink/ControlDesk in Verbindung mit der Hardware DS1104 für die Entwicklung mechatronischer Antriebssysteme vorgestellt. Diese Lösung basiert auf Open Source Ansätzen und eines speziell entwickelten mobilen Labors, welches die Studierenden ausleihen und zu Hause aufbauen und durchführen können. Am Beispiel eines rotatorischen Bedienelementes, angetrieben über einen Schrittmotor zur Abbildung einer mechatronischen Raste, wird zunächst die Modellbildung und anschließend die feldorientierte Regelung dargestellt. Am Ende können die Studierenden das Ergebnis ihrer Entwicklung haptisch erleben.

Einleitung

In der ingenieurwissenschaftlichen Ausbildung von Studierenden sind Labore, in der die theoretisch erworbenen Kenntnisse praxisnah angewendet werden, elementarer Bestandteil.

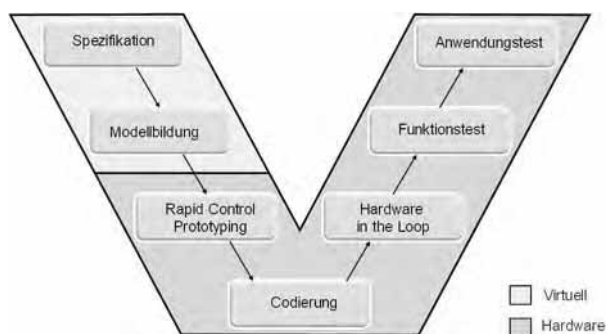


Abbildung 1: Entwicklungsprozess mechatronischer Antriebssysteme, unterteilt in virtuell und Hardware

Die Entwicklung mechatronischer Antriebssysteme folgt häufig einem V-Modell [1]. Die Umsetzung des V-Modells im Labor kann in virtuelle und hardwarenahe Elemente unterteilt werden (Abbildung 1). Die Phasen Spezifikation und Modellbildung können in der Ausbildung problemlos virtuell und damit online durchgeführt werden. Hierbei kommt entweder das kommerzielle

Werkzeug Matlab/Simulink oder die Open Source Variante Scilab/XCOS zum Einsatz. Für die weiteren Phasen ist jedoch ein Hardware-Aufbau erforderlich, welcher üblicherweise über ein Labor zur Verfügung gestellt wird. An der Ostfalia steht hierzu u.a. ein Pool Raum mit 13 Arbeitsplätzen zur Verfügung. Jeder Arbeitsplatz ist ausgestattet mit einem PC, dem Werkzeug Matlab/Simulink der Firma Mathworks und einer RCP-Hardware DS1104 mit dem Werkzeug ControlDesk der Firma dSpace. Da die RCP-Hardware nur eine Signalschnittstelle zur Verfügung stellt, wurde zusätzlich eine Leistungs- und Sensorelektronik aufgebaut (Abbildung 2).

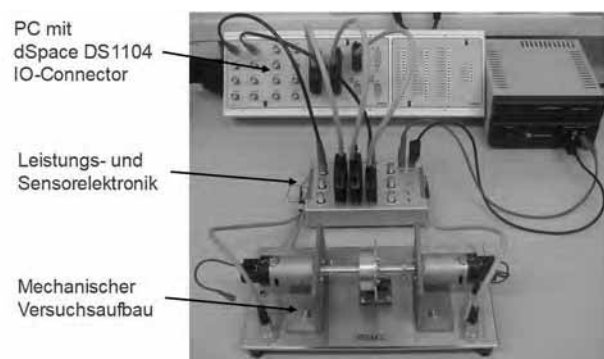


Abbildung 2: RCP-Labora Aufbau DC-Motor mit Leistungs- und Sensorelektronik zum Anschluss an DS1104

Die technischen Daten des Interface sind in Tabelle 1 aufgeführt. Hierbei ist anzumerken, dass die RCP-Hardware DS1104 nicht in der Lage ist, ein winkelabhängiges Einspritzsignal in Echtzeit zu erzeugen [2].

Eingänge	Ausgänge
1x 0...5V oder Potentiometer	1x 0...5V
1x Drehzahl (Zahnzeit) (Timer-Capture)	1x Einspritzsignal (Timer-Compare)
1x Encoder mit 5V Versorgung	2x PWM-Vollbrücke mit bis zu 100KHz, 12V, 2,5A mit bipolarer Strommessung

Table 1: Kenndaten der Leistungs- und Sensorelektronik

In dem Modul „Entwicklung mechatronischer Antriebssysteme“, welches vom Autor an der Ostfalia im

Bachelor-Studiengang Maschinenbau in der Vertiefungsrichtung Mechatronik und Digitalisierung im Umfang von 8 ECTS angeboten wird, erlernen die Studierenden anhand eines Gleichstrom- und eines, als Servomotor betriebenen Schrittmotors den Entwicklungsprozess. Als Prüfungsleistung müssen sie am Ende des Semesters an einem neuen Versuchsaufbau im Rahmen einer Projektarbeit die Phasen Spezifikation, Modellbildung und RCP-Implementierung einer Mehrgrößenregelung selbstständig durchführen. Hierzu stehen weitere Versuchsaufbauten zur Verfügung.

Bedingt durch die Corona-Pandemie, in der die Präsenzlabor gerade für größere Kohorten von Studierenden nur schwer genutzt werden können, fehlt ein zentrales Element in der Lehre. Daher wurde im Wintersemester 20/21 ein mobiles Labor aufgebaut, welches nachfolgend vorgestellt wird.

1 DS1104 und Alternative

Nach Vorstellung der RCP-Hardware DS1104 im Jahr 2002 [3] wurde diese in der Lehre und Forschung immer häufiger eingesetzt. So stieg die Anzahl der Veröffentlichung der IEEE-Datenbank, in der der Einsatz der DS1104 explizit erwähnt wurde, kontinuierlich an (Abbildung 3). Bereits 3 Jahre nach der Vorstellung der Hardware wurde diese auch an der Ostfalia in der Lehre und Forschung eingesetzt. Seit 2008 steht das in der Einleitung beschriebene Labor zur Verfügung.

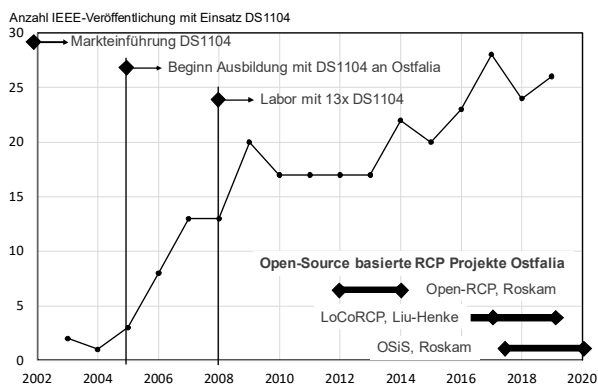


Abbildung 3: Historie Einsatz DS1104 und Projekte Ostfalia

Gerade in Zusammenarbeit mit kleinen und mittleren Industrieunternehmen musste in der Entwicklung festgestellt werden, dass die kommerzielle RCP-Hard- und Software häufig zu kostenintensiv ist. Daher begannen Entwicklungen, die auf Basis eines Open Source Ansatzes zu einer deutlich kostengünstigeren Lösung führen [4], [5]. So wurden auch am Institut für Mechatronik der

Ostfalia in der Vergangenheit Forschungsprojekte für Anwendungen der Hydraulik [6], Batterietechnik [7] und Sensorcluster [8] erfolgreich bearbeitet.

Insbesondere basierend auf den Vorarbeiten in [8] konnte innerhalb von 4 Monaten eine Alternative zum bisherigen Labor entwickelt werden. Diese Entwicklung nutzt einen Dual Core Mikrocontroller STM32H745.

	DS 1104	STM32H745
Vorstellung	2002	2019
Prozessor	Zwei Prozessoren 1x PPC603 1x TMS320F240	Dual Core STM32H745 1x Cortex M7 1x Cortex M4
Technologie	500nm	40nm
Takt	250 / 20 MHz	480 / 240 MHz
RAM/ROM	32M / 8M external	1M / 2M on chip
Floating Point Unit	1x Double precision 1x Keine FPU	1x Double precision 1x Single Precision
Timer	2x Encoder 10x PWM 4x Capture	6x Encoder 12x PWM 8x Capture/Compare
A/D	4 multiplexed, 16bit 4 parallel, 12 bit	20 multiplexed, 16bit 3 parallel, 16 bit
Interface	PCI, U(S)ART	USB, U(S)ART 2x CAN, Ethernet, SPI, I2C, SDMMC
Kosten	ca. 100	1
Werkzeuge	herstellerspezifisch	Open Source

Tabelle 2: Vergleich DS1104 [9] und STM32H745 [10]

Ein Vergleich der Hardware DS1104 mit dem Mikrocontroller zeigt, dass die dynamische Entwicklung in der Mikroelektronik der letzten 20 Jahre zu Lösungen geführt hat, die bessere Leistungsdaten (doppelte Taktrate, höhere Schnittstellenanzahl) zu erheblich geringeren Kosten realisiert. Da beim Einsatz einer DS1104 eine Leistungs- und Sensorelektronik als Interface aufgebaut werden muss, kann diese auch mit geringem Aufwand um einen Mikrocontroller erweitert werden.

Das Ergebnis der Hardware-Entwicklung ist das in Abbildung 4 dargestellte RESClab (Rapid Embedded Systems Control Laboratory). Das RESClab nutzt ein kostengünstiges Discovery-Board des Mikrocontrollers, welches zusätzlich über ein Touch-Display verfügt. Somit kann in der Weiterentwicklung zukünftig auch ein modernes HMI umgesetzt werden. Die Leistungs- und Sensorelektronik wurde in THT (Through-hole-technology) realisiert, so dass der Aufbau von Studierenden

selbst erstellt werden kann. Im Wintersemester 20/21 wurden von den Teilnehmenden des Moduls „Mechatronische Systementwicklung“ in 3 Doppelstunden zunächst 25 RESClabs zusammengebaut und in Betrieb genommen.



Abbildung 4: RESClab mit technischen Daten

2 RCP-Entwicklungswerkzeuge

Neben der Hardware sind durchgängige Software-Werkzeuge elementarer Bestandteil des mechatronischen Entwicklungsprozesses. Als Quasi-Standard hat sich in den letzten Jahren die Software Matlab/Simulink in der modellbasierten, mechatronischen Systementwicklung etabliert. Hiermit können Modelle und Regelalgorithmen blockorientiert entwickelt und automatisch in Quellcode und damit ausführbaren Programmen umgesetzt werden, die in Echtzeit auf einer speziellen Hardware ablaufen können. Um darüber hinaus während der Programmausführung Parameter ändern und Messgrößen aufzeichnen und analysieren zu können, wird für die DS1104 die Software ControlDesk der Firma dSpace eingesetzt.

Der Einsatz der durchgängigen, modellbasierten Werkzeugkette Matlab/Simulink/ControlDesk ist aus Kostengründen für den Einsatz in mobilen Laboren nicht geeignet. Daher wurde auf Basis des Forschungsprojektes OSiS [8] eine separate Werkzeugkette Scicos/XCOS/OSiS-IDE entwickelt, die ausschließlich mit Open Source Komponenten auskommt. Nach dem Start des Programms Scilab ist eine Auswahl zur gewünschten Hardware zu treffen, vergleichbar zur Hardware-Auswahl beim Start von Matlab in Verbindung mit RCP-Hardware der Firma dSpace. Anschließend stehen in Scilab/XCOS spezielle Hardware-Blöcke für das RESClab zur Verfügung. Im Hauptmenü kann nach Fertigstellung des Algorithmus in XCOS die automatische Code-Generierung über den OSiS-Codegenerator gestartet werden. Das Compilieren, Linken sowie der USB-Download des erstellten Programms erfolgt über eine selbstentwickelte,

integrierte Entwicklungsumgebung (OSiS-IDE). Hiermit können auch Parameter online geändert sowie Messgrößen erfasst und analysiert werden.

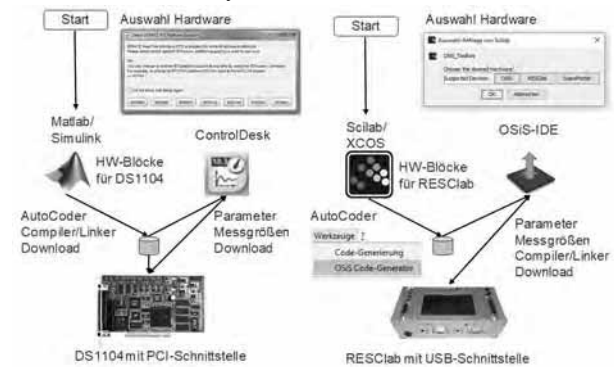


Abbildung 5: RCP-Entwicklungswerkzeuge im Vergleich

Der Funktionsumfang dieser Werkzeugkette ist deutlich geringer als bei der kommerziellen Variante. Allerdings ist die Funktionalität in den meisten Fällen und insbesondere in der Ausbildung des mechatronischen Entwicklungsprozesses vollkommen ausreichend, wie im nachfolgenden Kapitel gezeigt wird. Vorteilhaft ist dabei der, im Vergleich zur kommerziellen Werkzeugkette Matlab/Simulink/dSpace, geringere Einarbeitungsaufwand.

3 Einsatz RESClab

Nach Fertigstellung der Hard- und Software wurden in der zweiten Hälfte des Wintersemesters 20/21 die mobilen Labore verteilt. Das mobile Labor wird in einer Stapelbox (Größe 400x300x186mm) transportiert und umfasst das RESClab, unterschiedliche Versuchsaufbauten, Verbindungskabel, CAN-Bus mit Steckernetzteil 12V/5A sowie CAN-USB-Interface (Abbildung 6).

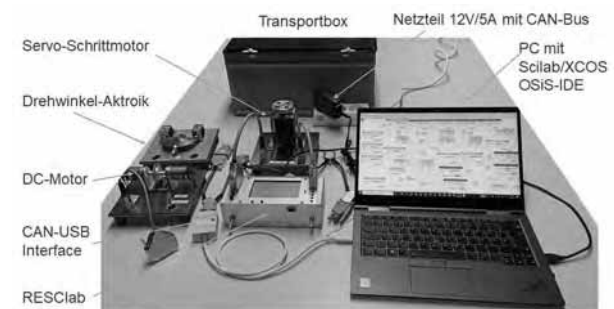


Abbildung 6: Mobiles Labor

Um das mobile Labor nutzen zu können, müssen die Studierenden einen PC oder Laptop mit Windows 7 oder 10 Betriebssystem und USB-Schnittstelle zur Verfügung stellen. Über einen Ausleihvertrag verpflichten sich die Studierenden zur Rückgabe der Geräte bis zum Ende des

Semesters.

Am Beispiel des Versuchsaufbaus Servo-Schrittmotor soll der Einsatz exemplarisch dargestellt werden. In der Spezifikation der Projektaufgabe wird eine Umsetzung eines rotatorischen Bedienelementes definiert, welches an einer einstellbaren Position über eine Raste verfügen soll (Abbildung 7).

Rotatorisches Bedienelement mit Rastfunktion

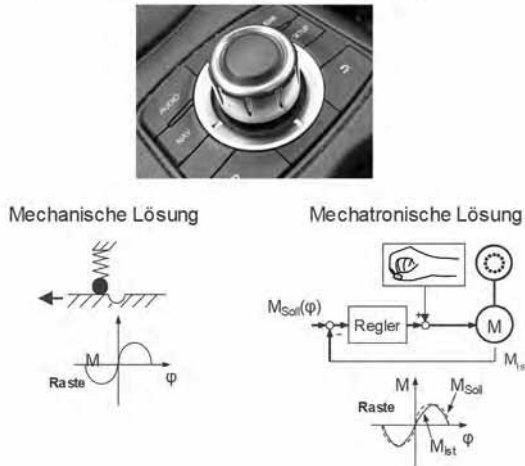


Abbildung 7: Rotatorisches Bedienelement mit Raste

Rastfunktionen werden mechanisch über eine feder vorgespannte Kugel realisiert, die an der gewünschten Position in einer Mulde einrastet. Um das Bedienelement aus der Raste zu bewegen, muss ein definiertes Moment überwunden werden. Diese Rastfunktion kann über einen Schrittmotor als mechatronische Raste umgesetzt werden [11].

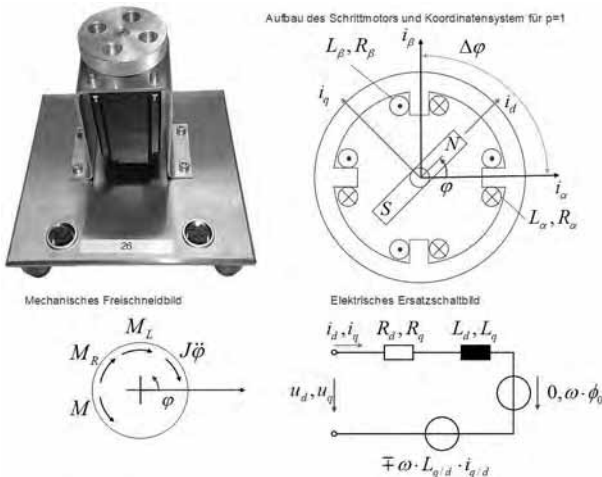


Abbildung 8: 2-Phasen Schrittmotor, Koordinatensystem und elektromechanische Ersatzschaltbilder

Der Schrittmotor ist dabei als Servoantrieb anzusteuern, der winkelabhängig ein definiertes Sollmoment ausregelt, welches der mechanischen Raste entspricht. Der Vorteil einer mechatronischen Realisierung ist die freie

Konfiguration der Rastposition und Haptik sowie auch die Umsetzung weiterer haptischer Rückmeldungen (Anschlag, Vibration, Reibung, usw.).

Nach der Spezifikation erfolgt die Modellbildung des Schrittmotors (Abbildung 8). Hierbei werden Reibmomente und Temperatureinflüsse vernachlässigt. Die mechanische Bewegungsgleichung des Produktes der Winkelbeschleunigung $\ddot{\varphi}$ und des Gesamtmassenträgheitsmomentes J des Rotors und der Drehscheibe ergibt sich aus der Summe der Momente:

$$J \cdot \ddot{\varphi} = M - M_L - M_R \quad (1)$$

Hierbei ist M das vom Schrittmotor erzeugte Moment, M_L das durch den Bediener wirkende Lastmoment und M_R das interne Rastmoment des Schrittmotors, welches über eine Sinusfunktion mit einer Amplitude M_{R0} winkelabhängig vereinfacht nachgebildet wird.

$$M_R = M_{R0} \cdot \sin(4 \cdot \vartheta) \quad (2)$$

Dabei ist zu berücksichtigen, dass der elektrische Winkel ϑ abhängig ist von der Polpaarzahl p des Motors und der aktuellen Winkelposition des Rotors φ .

$$\vartheta = p \cdot \varphi \quad (3)$$

Schrittmotoren besitzen üblicherweise eine Polpaarzahl von $p=50$, so dass sich im Schrittbetrieb als kleinster Winkelschritt ergibt:

$$\Delta\varphi = 360^\circ / (4 \cdot p) = 1,8^\circ \quad (4)$$

Das Motormoment M wird maßgeblich bestimmt durch den Strom i_q und der Motorkonstante ϕ_0 :

$$M = p \cdot (\phi_0 \cdot i_q + (L_d - L_q) \cdot i_d \cdot i_q) \quad (5)$$

Der Strom i_q ist der, in dem rotierenden d-q-Koordinatensystems quer zum Rotormagnetfeld wirkenden Strom, der mit Hilfe der Park-Transformation (6) aus den, in den beiden Spulen des Schrittmotors fließende Strömen i_α und i_β berechnet werden kann. Als Störgröße in der Momentengleichung wirkt die Differenz der, in das d-q-Koordinatensystem transformierten Induktivitäten L_d und L_q der beiden Spulen des Schrittmotors multipliziert mit den beiden Strömen i_d und i_q .

Die Modellbildung der elektrischen Kreise erfolgt im rotierenden d-q-Koordinatensystem des Rotors unter Verwendung der Park- und inversen Park-Transformation (7) in Bezug zum ortsfesten α - β -Koordinatensystem.

$$\begin{pmatrix} i_d \\ i_q \end{pmatrix} = \begin{pmatrix} \cos \varphi & \sin \varphi \\ -\sin \varphi & \cos \varphi \end{pmatrix} \cdot \begin{pmatrix} i_\alpha \\ i_\beta \end{pmatrix} \quad (6)$$

$$\begin{pmatrix} i_\alpha \\ i_\beta \end{pmatrix} = \begin{pmatrix} \cos \varphi & -\sin \varphi \\ \sin \varphi & \cos \varphi \end{pmatrix} \cdot \begin{pmatrix} i_d \\ i_q \end{pmatrix} \quad (7)$$

Ein Maschenumlauf ergibt die nachfolgenden Differenzialgleichungen:

$$u_d = R_d \cdot i_d + L_d \cdot \frac{di_d}{dt} - \omega \cdot L_q \cdot i_q \quad (8)$$

$$u_q = R_q \cdot i_q + L_q \cdot \frac{di_q}{dt} + \omega \cdot L_d \cdot i_d + \omega \cdot \phi_0 \quad (9)$$

Die Spannungen u_d und u_q sind die, in das d-q-Koordinatensystem transformierten Spannungen, die an den beiden Spulen des Schrittmotors angelegt werden. Die Berechnung erfolgt ebenfalls über die Park-Transformation. Neben dem PT1-Verhalten der beiden Stromkreise, welches über den jeweiligen Widerstand und Induktivität definiert ist, gibt es einen zusätzlichen Einfluss durch die jeweils andere Induktivität in Abhängigkeit der elektrischen Drehzahl ω und des, in dieser Induktivität fließenden Stroms. Die elektrische Drehzahl ω wird aus der Polpaarzahl und der mechanischen Drehzahl $\dot{\varphi}$ berechnet zu:

$$\omega = p \cdot \dot{\varphi} \quad (10)$$

In der q-Achse der Gleichung des Maschenumlaufs (9) muss zusätzlich die induzierte Spannung, die sich aus dem Produkt der elektrischen Drehzahl ω und der Motor-konstante ϕ_0 ergibt, berücksichtigt werden.

Typ: RTELLIGENT 42A03EC	Wert	Einheit
Haltemoment M_H	0,3	Nm
Schrittinkel / Polpaarzahl p	1,8 / 50	° / -
Nennstrom I_N	2	A
Massenträgheit Rotor J_M	77	gcm ²
Schrittzahl Encoder k	1000	-

Tabelle 3: Technische Daten des Schrittmotors [12]

Einige Parameter des Simulationsmodells können dem Datenblatt des Herstellers entnommen werden (Tabelle 3). Im Normbetrieb ist der Strom i_d Null, so dass sich die Motorkonstante ϕ_0 anhand der technischen Daten des Motors aus der Momentengleichung (5) berechnen lässt zu:

$$\phi_0 = M_H / (p \cdot I_N) = 3 \text{ mNm} / \text{A} \quad (11)$$

Zusätzlich erfolgen Messungen über das mobile Labor. Der Strom wird dabei über ein Anti-Aliasing-Filter mit einer Zeitkonstante T_{AA} von 0,12ms mit einer Abtastfrequenz von 5kHz gemessen. Der Winkel wird über einen Encoder mit einer 4-fach Auswertung und somit einer Auflösung von 4000 Inkrementen erfasst

Nach einmaliger Ausrichtung des Rotors durch Aktivierung einer Spule in der α -Achse, bei der gleichzeitig der gemessene Winkel des Encoders zu Null gesetzt wird, kann bei anschließend festgestelltem Rotor ($\omega=0$) unmittelbar die Sprungantwort des Stroms in der d-Achse (erneute Aktivierung der Spule in der α -Achse) und q-Achse (Aktivierung der Spule in der β -Achse) ermittelt werden (Abbildung 9).

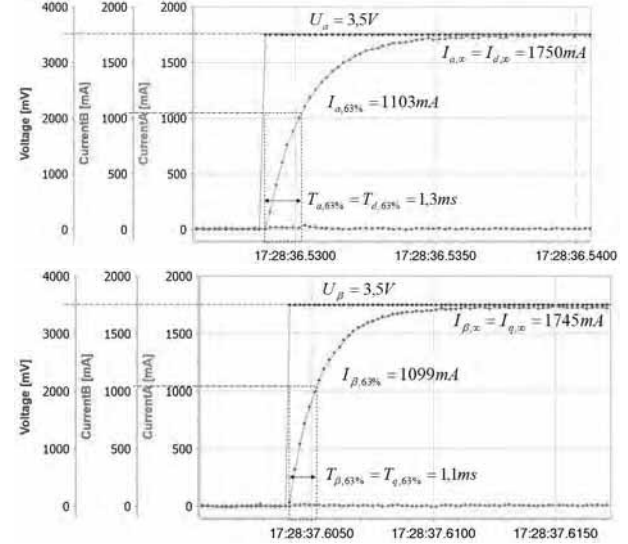


Abbildung 9: Sprungantwort Strom bei festgestelltem Rotor

Anschließend werden die Widerstände und Induktivitäten ermittelt. Die Sprungantwort des Stroms entspricht einem PT1-Verhalten, so dass die charakteristische Zeitkonstante bei Erreichen von 63% des Endwertes $T_{d/q,63\%}$ abgelesen werden kann. Um den Verzögerungseinfluss durch die Messung bei der Ermittlung der Induktivität zu berücksichtigen, wird die Zeitkonstante des Anti-Aliasing Filters T_{AA} vereinfacht subtrahiert:

$$R_{d/q} = U_{d/q} / I_{d/q,\infty} \quad (12)$$

$$L_{d/q} = (T_{d/q,63\%} - T_{AA}) \cdot R_{d/q} \quad (13)$$

Beim Massenträgheitsmoment muss neben dem Rotor die montierte Scheibe berücksichtigt werden. Dieses

wurde mit Hilfe der CAD-Daten zu $J_S=120\text{gcm}^2$ bestimmt. Ein Vergleich der Dynamik im Schrittbetrieb in Simulation und Messung zeigt eine gute Übereinstimmung, wenn das Massenträgheitsmoment 10% größer angesetzt wird (Abbildung 10),

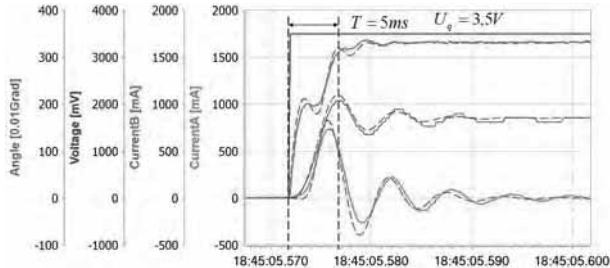


Abbildung 10: Validierung Schrittbetrieb (Simulation gestrichelt) mit angepasstem Massenträgheitsmoment

Zur Bestimmung des motorinternen Rastmomentes wird der Schrittmotor mit cosinus- und sinusförmigen Spannungen für U_α und U_β mit einer Amplitude von 3,5V und einer Frequenz ω von 50Hz angesteuert. Die Vorgabe einer Amplitude M_{R0} des Rastmomentes von 2,4% des Haltemomentes M_H führt dabei zu einer guten Übereinstimmung des Drehzahlverlaufs in Simulation und Praxis (Abbildung 11). Die Phasenverschiebung im Drehzahlverlauf ist bedingt durch die Zahnzeitmessung für nur einen Kanal des Encodersignals sowie eine Plausibilitätsüberprüfung (Erkennung einer Referenzmarke).

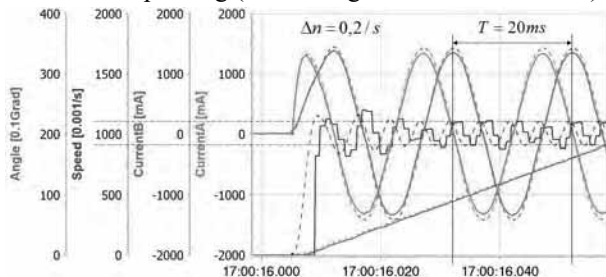


Abbildung 11: Validierung Sinusbetrieb (Simulation gestrichelt) mit angepasstem Rastmoment

Eine Übersicht der durch Messungen und mit Hilfe der Validierung bestimmten Parameter zeigt Tabelle 4.

Simulationsparameter	Wert	Einheit
Widerstand R_d	2,0	Ohm
Widerstand R_q	2,0	Ohm
Induktivität L_d	2,4	mH
Induktivität L_q	2,0	mH
Rastmoment M_{R0}	7,2	mNm
Massenträgheitsmoment J	217	gcm^2

Tabelle 4: Durch Messung und Validierung ermittelte Parameter des Schrittmotors

Nach Erstellung und Validierung der Simulation erfolgt die Auslegung des PI-Stromregelkreise sowohl für die d- als auch für die q-Achse (Abbildung 12) nach der Methode des Betragsoptimums einer PT2-Regelstrecke [13].

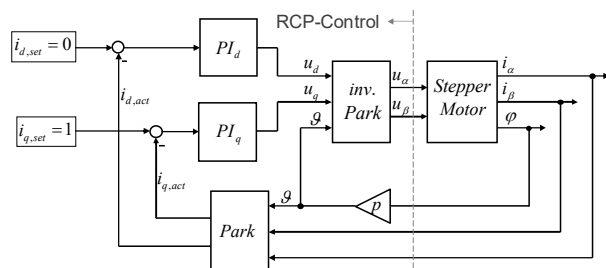


Abbildung 12: Stromregelkreise für d- und q-Achse

Die PT2-Glieder ergeben sich aus den beiden Zeitkonstanten der LR-Netzwerke sowie des Anti-Aliasing-Filters in der d- und q-Achse. Die Gesamtverstärkung des PT2-Gliedes wird über dem Widerstand des Netzwerkes definiert. Bei der Reglerauslegung wird die größte Zeitkonstante der Strecke (hier das LR-Netzwerk) kompensiert und für das PT2-Ersatzsystem des geschlossenen Regelkreises eine Dämpfung von 1 vorgegeben. Somit ergeben sich die Regelparameter in nichtnormierter Form zu:

Regelparameter	Wert	Einheit
d-Kreis Proportional $K_{p,d}$	4,9167	Ohm
d-Kreis Integral $K_{i,d}$	4167	Ohm/s
q-Kreis Proportional $K_{p,q}$	4,0950	Ohm
q-Kreis Integral $K_{i,q}$	4179	Ohm/s

Tabelle 5: PI-Regelfaktoren gemäß Betragsoptimum

In der Lehre wird im Modul „Mechatronische Systementwicklung“ bei der Auslegung der Regelung zusätzlich auf eine drehzahlabhängige Vorsteuerung zur Kompensation der Induktionsstöreinflüsse des PT1-Gliedes im Stromkreis, auf die Limitierung der Regelung durch Abtastung und Diskretisierung der Aktor- und Sensorgößen, sowie auf die Begrenzung der Stellgröße und den dadurch erforderlichen Anti-Wind-Up Regler eingegangen. Die Analyse erfolgt dabei zunächst in der Simulation und anschließend mit Hilfe des mobilen Labors am realen Versuchsaufbau.

Hier wird im Folgenden lediglich die Stromregelung ohne Vorsteuerung dargestellt. Mit den analytisch berechneten Regelparametern wird ein Sollwertsprung auf 1A in der d-Achse sowohl in der Simulation als auch in der Praxis nach ca. 1ms ohne Überschwinger ausgeregelt (Abbildung 13). Bei Vorgabe des gleichen Sollwertsprungs in der q-Achse verbleibt eine Regelabweichung

(Abbildung 14). Diese kann mit einer Vorsteuerung der drehzahlabhängigen Induktionsstöreinflüsse des PT1-Gliedes im Stromkreis vollständig ausgeglichen werden, setzt aber eine zusätzliche Messtechnik oder einen Beobachtungsalgorithmus für die Drehzahl voraus. Da die Drehzahl in der Anwendung als mechatronische Raste bei manueller Bedienung üblicherweise klein ist, wird die verbleibende Regelabweichung deutlich kleiner ausfallen, so dass eine Vorsteuerung entfallen kann.

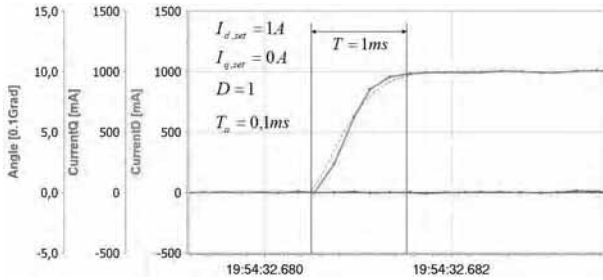


Abbildung 13: Ergebnisse der Stromregelung in der d-Achse ohne Bewegung (Simulation gestrichelt)

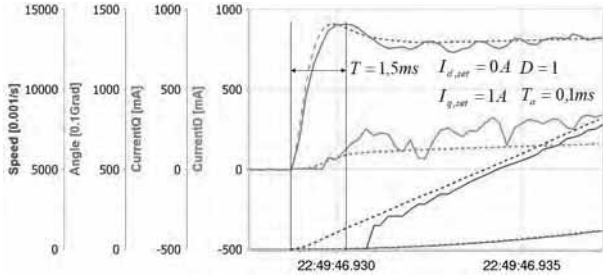


Abbildung 14: Ergebnisse der Stromregelung in der q-Achse mit Bewegung (Simulation gestrichelt)

Um eine Rastfunktion abzubilden, muss eine winkelabhängige Momentvorgabe realisiert werden. Hierzu können unterschiedliche Verläufe definiert werden, die die Qualität der haptische Rückmeldung bestimmen. Eine gute Haptik wird gemäß [11] durch eine Sinusfunktion des Sollmomentes erreicht (Abbildung 15).

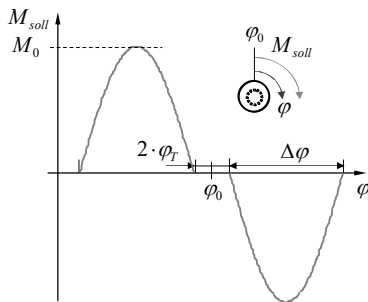


Abbildung 15: Stromsollwert einer Raste mit Nullpunkt

Da das Moment näherungsweise proportional zum Strom i_q ist, ist für die haptische Rückmeldung eine winkelabhängige Sollwertvorgabe des Stromes ausreichend. Hierzu wird eine sinusförmige Stromsollwertvorgabe mit der Amplitude I_0 definiert. Der Stromsollwert ergibt sich

zu:

$$\text{für } \varphi_0 + \varphi_T < \varphi < \varphi_0 + \varphi_T + \Delta\varphi:$$

$$i_{q,soll} = I_0 \cdot \sin\left(\frac{\pi}{\Delta\varphi} \cdot (\varphi - (\varphi_0 + \varphi_T))\right)$$

$$\text{für } \varphi_0 - \varphi_T > \varphi > \varphi_0 - \varphi_T - \Delta\varphi:$$

(14)

$$i_{q,soll} = I_0 \cdot \sin\left(\frac{\pi}{\Delta\varphi} \cdot (\varphi - (\varphi_0 - \varphi_T))\right)$$

$$\text{sonst: } i_{q,soll} = 0$$

Hierbei ist φ_0 die gewünschte Rastposition, φ_T eine symmetrische Breite um die Rastposition, in der kein Moment wirkt und $\Delta\varphi$ die Breite, auf der eine positive Halbwelle der Sinusfunktion wirken soll. Der Strom i_d wird zu Null vorgegeben.

Die Umsetzung des Stromsollwertgenerators erfolgt in einem Superblock in XCOS mit Parametern, die über die OSiS-IDE online einstellbar sind (Abbildung 16).

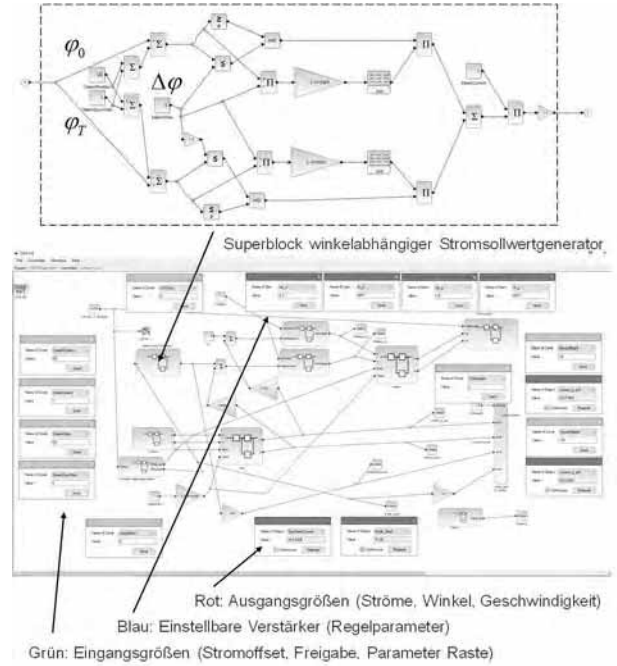


Abbildung 16: Stromsollwertgenerator und OSiS-IDE

Abschließend kann die haptische Rückmeldung der Raste, die durch die Regelung erzeugt wird, vom Bediener mit Hilfe des mobilen Labors „gefühl“ werden. Mit Hilfe der einstellbaren Parameter ist eine Optimierung der Haptik möglich. Exemplarisch wurden die in der Tabelle 6 angegebenen Parameter gewählt. Gerade durch das reale Erleben der Regelung einer mechatronischen Raste in der Praxis zeigen sich sehr deutlich die Grenzen der virtuellen Ausbildung und die Vorteile eines mobilen Labors. Auch in dieser Veröffentlichung kann nur das Messergebnis der Prozessgrößen beim Durchfahren der

mechatronischen Raste dargestellt (Abbildung 17), nicht jedoch die Haptik erlebt werden.

Parameter der Raste	Wert	Einheit
Amplitude Raststrom I_0	1	A
Rastposition φ_0	90	Grad
Rastbreite $\Delta\varphi$	15	Grad
Breite Nullpunkt φ_T	1	Grad

Tabelle 6: Parameter der Raste

Durch die manuelle Bedienung des Stellelementes ist der Winkelverlauf nichtlinear ansteigend. Deutlich ist zu erkennen, dass die Regelung des Stromes jederzeit in der Lage ist, den Istwert ohne Abweichungen dem Sollwert folgen zu lassen.

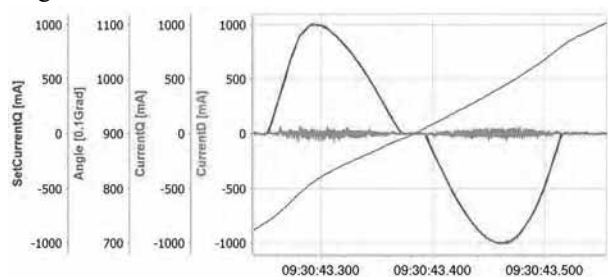


Abbildung 17: Prozessgrößen beim manuellen Durchfahren einer mechatronischen Raste

Die für die Anwendung mechatronische Raste von der OSiS-IDE berechnete Auslastung des Prozessorkerns M7 beträgt, bei einer Abtastzeit von 0,2ms ohne den Einsatz von Optimierungen, maximal 47%. Aufgrund der begrenzten Entwicklungszeit wurde der Prozessorkern M4 noch nicht genutzt. In der Weiterentwicklung des RESClabs ist somit noch eine deutliche Leistungssteigerung möglich. Der Speicherplatzbedarf des automatisch erzeugten Programms umfasst ca. 92KByte inkl. IO-Handling sowie CAN-Bus und USB-Kommunikation, so dass der interne Speicher des Mikrocontrollers von 2MByte vollkommen ausreichend ist.

4 Zusammenfassung

In dieser Veröffentlichung wurde ein mobiles Labor auf Open Source Basis vorgestellt, welches auf einem neuen Mikrocontroller basiert. Aufgrund der dynamischen Entwicklung in der Mikroelektronik ist dieses Labor von den Leistungsdaten vergleichbar zu der, in der Entwicklung elektrischer Antriebssysteme häufig eingesetzten Hardware DS1104 der Firma dSpace mit den

Software-Werkzeugen Matlab/Simulink/ControlDesk.

Anhand der Projektaufgabe „Mechatronische Raste“ konnten sowohl Messungen zur Parameterermittlung wie auch die Umsetzung einer feldorientierten Stromregelung dargestellt werden. Das Ergebnis der mechatronischen Raste kann durch eine Messung validiert werden, jedoch nur durch das reale Fühlen der Haptik mit Hilfe des mobilen Labors von den Studierenden erlebt werden.

References

- [1] Verein Deutscher Ingenieure (Hrsg.). VDI 2206 - Entwicklungsmethodik für mechatronische Systeme. Berlin, Beuth Verlag GmbH 2004
- [2] Roskam, R. Mechatronic Evaluation Board for Use in Education of the Mechatronic Engineering Process. 11th International Workshop on Research and Education in Mechatronics, Ostrava, 2010
- [3] Pöhlmann, T. Antriebsregler leicht gemacht. Embedded engineering, München, Hanser 2002
- [4] Bucher, R.; Balemi, S. Scilab/Scicos and Linux RTAI - A unified approach. IEEE Conference on Control Applications, Toronto, 2006
- [5] Grzegorz et al. Rapid Control Prototyping with Scilab/Scicos/RTAI for PC-based and ARM-based Platforms. Proceedings of the IMCSIT Volume 3, Wisla, Poland, 2008
- [6] Dobkowitz, D. et al. Serientauglicher Hydraulikcontroller auf Open Source Basis für die modellbasierte Entwicklung hydraulischer Systeme. 8. Kolloquium Mobilhydraulik, Braunschweig 2014
- [7] Jacobitz, S; Liu-Henke, X. The Seamless Low-cost Development Platform LoRra for Model based Systems Engineering. Proceedings of the 8th International Conference on Model-Driven Engineering and Software Development, Valetta 2020
- [8] Roskam, R. et al. Abschlussbericht zum Forschungsvorhaben .ZW 6- 85007817 OSiS - Offene Steuerung mit integriertem Sensorcluster. Ostfalia 2020
- [9] NN.: Product Information DS1104. Paderborn, dSpace 2020
- [10] N.N.: Reference Manual RM0399 STM32H745/755 and STM32H747/757 advanced Arm®-based 32-bit MCUs V3.0, STMicroelectronics, 2020
- [11] Roskam, R. Abschlussbericht zum BMBF Forschungsvorhaben 03FH049PX2 "Integratives Werkzeug zur Entwicklung haptischer Bedienelemente", Ostfalia 2016
- [12] N.N. Datenblatt Servo Stepping Motor 42A03EC. Rtelligent, 2020
- [13] Schröder, D. Elektrische Antriebe - Regelung von Antriebssystemen. Berlin, Springer-Verlag 2015

Qualification Considerations for Simulations in Avionics Software Engineering

Mohamad Ibrahim^{1*}, Umut Durak²

¹Institute of Informatics, TU Clausthal, Julius-Albert-Str. 4,
38678 Clausthal-Zellerfeld, Germany; **mohamad.ibrahim@tu-clausthal.de*

²Institute of Flight Systems, German Aerospace Center (DLR), Lilienthalplatz 7, 38108 Braunschweig, Germany

Abstract. Simulation engineering has become an established practice in software development processes in various domains. ARP4754A Guidelines for Development of Civil Aircraft and Systems and DO-178C Software Considerations in Airborne Systems and Equipment Certification have adopted simulation to aid in requirements validation and implementation verification. Tool qualification requirements have already been settled in safety-critical domains. However, the methods and guidelines for applying these requirements in the simulation engineering life cycle are still missing. This work introduces the background in tool qualification, presents the state of the art, and discusses the qualification considerations for simulation in avionics software engineering.

Introduction

Simulation engineering is performed in the process of software development for different domains. It is a method that supports the verification and validation of software systems and helps to uncover and assess risk factors. Simulation engineering is the execution of an interdisciplinary systems engineering process for developing, maintaining, and employing simulations, which enable systems engineers to experiment and gain insight into the systems of interest [1].

Simulation has been widely adopted in safety-critical software development. ARP4754A Guidelines for Development of Civil Aircraft and Systems referred to the use of simulation to achieve requirements validation and implementation verification. Requirements validation ensures correctness and completeness to meet the stakeholders' needs such as flight crew or developers. While implementation verification ensures the conformance of the system implementation to its validated requirements.

In the avionic domain, developers use simulation with different software artifacts, namely, Specification Models, Design Models, and executable code. DO-331 [2], the model-based development and verification supplement to DO-178C promotes simulation as a way to satisfy

its objectives [3]. DO-331 accepts simulation as means to satisfy the verification objectives of the specification and design models, however, it accepts simulation only in combination with testing and coverage analysis to satisfy the verification objectives of the executable code.

This practice of using simulation to satisfy certification or development assurance objectives makes the correctness of simulation a safety concern. Besides, as the complexity of the system increases, its simulation also becomes remarkably complex, that it is necessary to assess it, not only as a valid support for systems engineering processes but also as an objective of systems engineering efforts. One approach to address this safety concern is through the adoption of tool qualification as a framework for quality assurance.

Tool qualification is "a process that allows us to demonstrate that a tool can be used as part of the realization of a software application with a determined safety goal"[4]. Tool qualification guarantees that the tool is developed and verified using an adequate process to obtain confidence in the tool's functionality.

IEEE 1730 Distributed Simulation Engineering and Execution Process (DSEEP) is a recommended practice for simulation life-cycle processes. In conjugation with DO-330 [5], the Software Tool Qualification Considerations supplement to the avionics standards RTCA DO-178C, DSEEP can form a basis for a simulation qualification strategy.

Structure of the paper. Section 1 is a summarization of the most important aspects of tool qualification and then focuses on the DO-330 guidelines. In section 2, we discuss the current utilization of simulation engineering in the avionic system and software development. Section 3 is dedicated to paving the floor to the proposed approach. Finally, we conclude in section 4.

1 Tool Qualification

Tool qualification is the process of documenting pieces of evidence that show the tool is reliable and fit the intended purpose in the context of a specific use case. Boulanger defines tool qualification in [4] as “a process that allows us to demonstrate that a tool can be used as part of the realization of a software application with a determined safety goal”.

Software tools reside in two categories [6], Software Development Tools and Software Verification Tools. The categories contrast in which the development tools can insert an error in the software while verification tools can only fail to detect an error in the software [6]. Examples of Development Tools are compilers, linkers, modeling tools, code generators, code manipulators, etc. Verification Tools include test case generators, code static analysis, test automation, structural coverage tools, test results checker, etc.

All safety standards adopt the principle that says: “qualification of the tool is required only when this tool replaces, reduces, or automates one of the software life-cycle processes”. Nevertheless, the qualification process can be eliminated in case the output of the tool is verified by another process or a qualified tool. In other words, if the activities or tasks required by a standard rely on the correct functioning of the tool, then tool qualification is a necessity [7].

In general, there are four methods/approaches for tool qualification that is accepted by most standards [8]:

1. Increased confidence from use (proven in use argumentation)
2. Evaluation of the development process (process assessment)
3. Validation of the software tool in the operational environment
4. Development in compliance with a safety standard

The avionic standard for functional safety and certification aspects DO-178C and its tool qualification supplement DO-330 only accept the last two methods. The 3rd method represents TQL-5 (the lowest Qualification Level of DO-330), and the fourth method consists of presenting evidence of developing the tool according to a safety standard such as DO-178C, and thus it involves input from the developer. The rest of this section will focus only on

the qualification process of the DO-178C/DO-330. This brief introduction to DO-330 is adopted from our previous work [9].

1.1 Tool Classification and Analysis

All qualification methods start with evaluating the tool’s impact on the process workflow of the software lifecycle. DO-178C defines three criteria (Tool Impact) for the Tool under Qualification:

Table 1: DO-178C Tool Impact Criteria [5]

a.	Criteria 1: A tool whose output is part of the airborne software and thus could insert an error.
b.	Criteria 2: A tool that automates verification process(es) and thus could fail to detect an error, and whose output is used to justify the elimination or reduction of: <ol style="list-style-type: none"> 1. Verification process(es) other than that automated by the tool, or 2. Development process(es) that could have an impact on the airborne software.
c.	Criteria 3: A tool that, within the scope of its intended use, could fail to detect an error.

Criteria 1 corresponds to the Development Tools category, while Criteria 2 and 3 represents verification tools. Criteria 3 tools should be extended to Criteria 2 if they are used for eliminating or reducing a process that is mandated to be used by DO-178C in the software life cycle.

Based on the criteria of the tool, finding the TQL is a matter of table lookup activity. The following table determines the TQL

Table 2: Tool Qualification Level Determination

Software Level	Criteria		
	1	2	3
A	TQL-1	TQL-4	TQL-5
B	TQL-2	TQL-4	TQL-5
C	TQL-3	TQL-5	TQL-5
D	TQL-4	TQL-5	TQL-5

Based on the resulted target TQL, DO-330 specifies the set of objectives that the qualifier should fulfill to achieve the qualification of the tool. These objectives are set in processes that constitute the life cycle of the tool qualification.

1.2 Qualification Life Cycle and Processes

Software quality cannot be added to a product after it is developed [6]. Thus, DO-330 defines tool qualification

life cycle processes (Fig.1 adopted from [10]) to qualify the tools and meet the required quality and assurance:

1. Tool Operational Process
2. Tool Planning Process
3. Tool Development Process
4. Tool Verification Process
5. Integral Processes is done throughout the entire tool qualification life cycle.

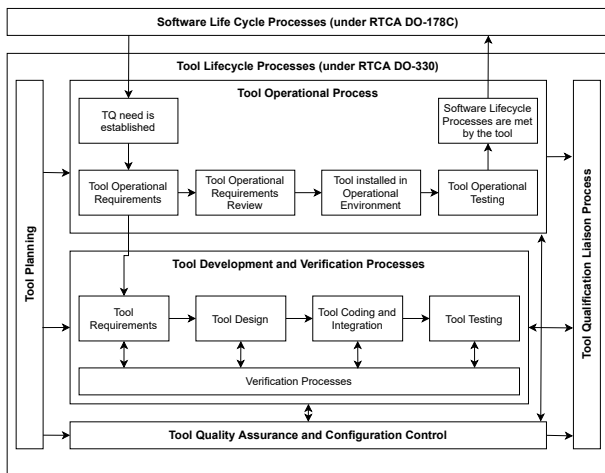


Figure 1: Tool Qualification Life Cycle

Tool Operational Process is the user's responsibility. The main objectives involved in this process include: tool qualification need is established, TOR (Tool Operational Requirements), tool executable code installed, and TOR is validated [10].

Tool Planning Process defines the tool qualification processes and their interaction/interrelationship for the tool qualification life cycle. In the planning process, standards, verification environment, and development environment should be defined. Lastly, the qualifier should enumerate the DO-330 objectives and describe how they will be fulfilled.

Tool Development Process is the implementation process of the tool. High-Level Requirements (HLR), Low-Level Requirements (LLR), and code are developed considering traceability, transition, and integration criteria in the process.

Tool Verification Processes is done sequentially in two phases, the first is the verification of the tool functional requirements in which the tool developers' work needs to be verified against the intended functionality (Fig. 1 the "Tool Development and Verification Processes" block). Secondly, is the tool operational verification and validation process in which the tool user's work needs to be verified against the intended usage (Fig. 1 the "Tool Operational Process" block).

2 Safety-critical Simulation Engineering in Airborne Development

Guidance for the development of airborne systems for civil aircraft can be found in ARP4754A [11]. It defines the development process as a set of activities from the early stages of conceptualization to certification final phases. ARP4754A emphasizes the simulation sufficiency, appropriateness, and validation. Notwithstanding, there is no clear definition of the simulation process nor its quality assurance requirements and objectives. DO-178C supports ARP4754A by providing process requirements for the entire software development life cycle. DO-178C and its supplements allude to simulation in two contexts:

1. **Model Simulation Context:** refers to the execution of the Specification or Design Model to collect pieces of evidence about the compliance of Specification Models to system requirements, compliance of Design Models to software high-level requirements, and further model qualities such as accuracy or consistency.
2. **Executable Simulation context:** it aims to reveal errors that usually arise from running the software on the target hardware. Such use mandates to prove that the simulator is representative of the actual target hardware.

The utilization of simulation requires a simulation tool or simulator. DO-178C requires in some cases the simulator to be qualified based on its use. The qualification process is well-defined in the DO-178C supplement DO-330. Thus, our work focuses on the quality of the simulation process itself and not on the quality of the simulation tool or simulator. Furthermore, for the sack of

showcasing, we will be proposing an approach that addresses the model simulation context, in which the objectives are confined to the verification and validation of the design models.

The problem of verifying the correctness of the simulation models has long been one of the major concerns of the simulation engineering community in model-based software development. It dates back to the 1960s. A survey that addresses the work on this topic since the early days is made by Sargent et al. [12]. The subsequent research and academic efforts have led to the standardization of the modeling and simulation life cycle as part of the High-Level Architecture (HLA) standard suite. IEEE Std 1516.3-2003, IEEE Recommended Practice for High-Level Architecture (HLA) Federation Development and Execution Process (FEDEP), proposed a process for federation development, particularly for distributed simulations that utilize HLA [13].

FEDEP was then generalized by the Simulation Interoperability Standards Organization (SISO) FEDEP Product Development Group (PDG) to support the engineering process for all types of distributed simulation. This was published in 2010 as IEEE Std 1730-2010: IEEE Recommended Practice for Distributed Simulation Engineering and Execution Process (DSEEP) [14]. With DSEEP we have got a strong basis to build our approach for the simulation qualification since it is currently the de facto standard and the most well-received standard process for simulation engineering.

DSEEP is a generalized process for simulation engineering [14]. Although it refers to distributed simulation engineering, it is also applicable for stand-alone applications. It presents the best practices through defining the processes and procedures to be followed to develop and execute simulations, following is the flow of the DSEEP processes:

- *Step 1: Define Simulation Environment Objectives* aims at specifying the needs and eventually objectives that are to be addressed with the simulation to be developed and executed.
- *Step 2: Perform Conceptual Analysis* aims at modeling, scenario development, and requirements specification.
- *Step 3: Design Simulation Environment* includes identifying member applications, reuse candidates, and planning.

- *Step 4: Develop Simulation Environment* includes developing data exchange models, simulation environment agreements, member application development, and infrastructure implementation.
- *Step 5: Integrate and Test Simulation Environment* aims at integrating all the member applications using the implemented infrastructure and testing the environment before execution.
- *Step 6: Execute Simulation* is the step where the simulation (all member applications) is executed and the results are collected.
- *Step 7: Analyze Data and Evaluate Results* aims at the analysis and the evaluation of the collected data

Similar to DO-330, each step in DSEEP is further branched to activities. Furthermore, each activity is concertized to activity input, activity output, and recommended tasks.

3 Simulation Qualification

This paper presents an extension to the previous papers [15, 16, 17]. In this paper, the authors try to promote and plant the seeds for new practices that establish Simulation Qualification as a requirement in the avionics software development domain. Thus, we propose the utilization of DO-330 and DSEEP as the foundations for simulation qualification. The approach takes guidance from the qualification approach in the airborne system's domain -specifically DO-330- to achieve objectives mandated by DSEEP. This will facilitate its usage among DO-178C practitioners and experts and make it more acceptable and complied with the industry's high standards.

We propose not to update, enhance or push DSEEP more towards DO-330 [17, 16], but rather get DSEEP as a self-contained and complete simulation engineering life cycle process and trace its recommendations to the requirements for the qualification of simulations for safety-critical systems engineering [15].

Despite the structure of DO-330 and DSEEP are different, Both guidelines will fill in different parts of the proposed approach. E.g. the integral processes like verification and validation or configuration management are not explicit, but referred to and explained within the steps of DSEEP.

The approach initiates similar to DO-330, by evaluating the simulation impact as follow:

- Criteria 1:** A simulation whose output is part of the system and thus can introduce an error
- Criteria 2:** A simulation that is used in verification and validation of the system and this could fail to detect an error, and whose output is used to justify the elimination or reduction of another validation and verification effort.
- Criteria 3:** A simulation, within the scope of its intended use, could fail to detect an error.

Subsequently, we build on the elected criteria to determine the SQL (Simulation Qualification Level), the lookup table is depicted in Table 3:

Table 3: Determination of SQL.

Criteria	Development Assurance Level			
	A	B	C	D
1	SQL1	SQL2	SQL3	SQL4
2	SQL4	SQL4	SQL5	SQL5
3	SQL5	SQL5	SQL5	SQL5

DO-330 describes the objectives as requirements that need to be fulfilled to demonstrate compliance with the document. We propose to adopt the DSEEP activities as the major objectives and enhance them with the inline statements in activity definitions regarding the support processes, or integral processes, such as verification and validation. As a sample we make an overlay for the DSEEP Perform conceptual analysis process, Table 4 depicts the objectives for this process.

Table 4: Perform Conceptual Analysis Step Objectives.

Objective	Ref.	Recommended Tasks	Applicability by SQL					Output	Ref.	Control Category by SQL				
			1	2	3	4	5			1	2	3	4	5
Develop scenario	4.2.1	4.2.1.2	○	○	○	○	○	Scenario(s)	4.2.1.3	①	①	①	①	①
Develop conceptual model	4.2.2	4.2.2.2	○	○	○	○	○	Conceptual Model	4.2.2.3	①	①	①	①	①
Develop simulation environment requirements	4.2.3	4.2.3.2	○	○	○	○	○	Simulation Environment Requirements Simulation Environment Test Criteria	4.3.3.3	①	①	①	①	①

The objectives are listed as: Develop scenario, Develop conceptual model, and Develop simulation environment requirements. The reference for the objective points to the related section in DSEEP that explains the activity. The recommended Tasks replace the Activity references in DO-330; they are the means to meet the objectives. Aligned with the requirements engineering objectives of DO-330, the objectives apply to all SQLs from 1 to 4. The Outputs refer to DSEEP activity outputs. For example, as described in section 4.3.3.3 of DSEEP, Simulation Environment Requirements and Simulation Environment Test Criteria are the outputs of the Develop Simulation Environment objective. The control categories for all objectives are assigned as 1.

The objective tables of DO-330 are organized in such a way that there is a table for the verification of outputs of each development process. Accordingly, Table 5 lists the objectives for the verification of the outputs of Perform Conceptual Analysis step.

Table 5: Verification of Conceptual Analysis.

Objective	Ref.	Recommended Tasks	Applicability by SQL					Output	Ref.	Control Category by SQL				
			1	2	3	4	5			1	2	3	4	5
Scenarios include major entities, their capabilities, behavior and relations among each other, environmental, initial and terminal conditions.	4.2.1	4.2.1.2	●	●	○	○	○	As specified in V&V plan	4.1.3.3	②	②	②	②	②
Conceptual model represents the domain adequately.	4.2.2	4.2.2.2	●	●	○	○	○	As specified in V&V plan	4.1.3.3	②	②	②	②	②
Requirements comply with objectives statement.	4.2.3	4.2.3.2	●	●	○	○	○	As specified in V&V plan	4.1.3.3	②	②	②	②	②
Requirements are testable.	4.2.3	4.2.3.2	●	●	○	○	○	As specified in V&V plan	4.1.3.3	②	②	②	②	②
Requirements give implementation level guidance.	4.2.3	4.2.3.2	●	●	○	○	○	As specified in V&V plan	4.1.3.3	②	②	②	②	②
Requirements address execution management	4.2.3	4.2.3.2	●	●	○	○	○	As specified in V&V plan	4.1.3.3	②	②	②	②	②
Requirements explicitly address fidelity	4.2.3	4.2.3.2	●	●	○	○	○	As specified in V&V plan	4.1.3.3	②	②	②	②	②

The objectives are based on the text that describes the corresponding activity in DSEEP. Simply explained, “shall” statements are reworded as verification objectives. An example would be “Requirements comply with the objectives statement”. The objective and the recommended tasks can be found in sections 4.2.3 and 4.2.3.2

of DSEEP, respectively. The objective shall be fulfilled for the SQLs 1 to 4, and with independence, designated by the solid dot, for the SQL-1 and SQL-2.

4 Conclusion and Outlook

The paper presents an overview of the Tool Qualification process and a brief overview of the simulation engineering processes in the avionic domain. Upon that, it constructs an approach to establish a framework for the qualification of simulations to be used in safety-critical systems engineering. The proposed framework extends DO-330 with simulation processes and objectives from DSEEP to achieve quality assurance in the model simulation process. It is important to note that the paper relies on DSEEP as a self-contained and complete process framework for simulation engineering. It proposes to augment it with definitions of Simulation Qualification Levels (SQLs) and a set of objectives that are required to achieve these SQLs. This approach forms a firm ground for future research in the quality assurance of the simulation process under safety-critical software development. The further research aims to extend best practices in simulation engineering in a widely accepted guideline document that makes simulation a more reliable means of standard compliance.

References

- [1] Durak U, Ören T. Towards an ontology for simulation systems engineering. In: *Proceedings of the 49th Annual Simulation Symposium*, ANSS '16. Pasadena, California: Society for Computer Simulation International. 2016; pp. 1–8.
- [2] RTCA. DO-331 The model-based development and verification supplement to DO-178C. *Radio Technical Commission for Aeronautics*. 2012;.
- [3] RTCA. DO-178C Software Considerations in Airborne Systems and Equipment Certification. *Radio Technical Commission for Aeronautics*. 2011;.
- [4] Boulanger JL. *Certifiable Software Applications 2: Support Processes*. Elsevier. 2017.
- [5] RTCA. DO-330 Software Tool Qualification and Considerations. *Radio Technical Commission for Aeronautics*. 2011;.
- [6] Hilderman V. DO-330: Tool Qualification Overview for Avionics Engineers and Managers. *www.afuzion.com*. 2017;.
- [7] Conrad M, Sandmann G, Munier P. Software Tool Qualification According to ISO 26262. In: *SAE 2011 World Congress Exhibition*. SAE International. 2011; .
- [8] ISO. 26262: Road vehicles-Functional safety. *International Standard ISO/FDIS*. 2011;26262.
- [9] Ibrahim M, Durak U. State of the Art in Software Tool Qualification with DO-330: A Survey. In: *Proceedings of the Software Engineering 2021 Satellite Events, Braunschweig/Virtual, Germany, February 22 - 26, 2021*, edited by Götz S, Linsbauer L, Schaefer I, Wortmann A, vol. 2814 of *CEUR Workshop Proceedings*. CEUR-WS.org. 2021; .
- [10] Marques J, Marques da Cunha A. COTS tool qualification using RTCA DO-330: Common pitfalls. In: *2017 IEEE/AIAA 36th Digital Avionics Systems Conference (DASC)*. 2017; pp. 1–6.
- [11] SAE. ARP4754A Guidelines for Development of Civil Aircraft and Systems. *SAE International*. 2010;.
- [12] Sargent RG, Balci O. History of verification and validation of simulation models. In: *2017 Winter Simulation Conference (WSC)*. IEEE. 2017; pp. 292–307.
- [13] IEEE. IEEE Recommended Practice for High Level Architecture (HLA) Federation Development and Execution Process (FEDEP). *IEEE Std 15163-2003*. 2003;.
- [14] IEEE. IEEE Recommended Practice for Distributed Simulation Engineering and Execution Process (DSEEP). *IEEE Std 1730-2010*. 2011;.
- [15] Durak U, D'Ambrogio A, Bocciarelli P. Safety-Critical Simulation Engineering. In: *Proceedings of the 2020 Summer Simulation Conference*, Summer-Sim '20. San Diego, CA, USA: Society for Computer Simulation International. 2020; .
- [16] Durak U, D'Ambrogio A, Gerlach T. Applying IEEE Recommended Practice for Distributed Simulation Engineering and Execution Process for Mod-

eling and Simulation Based Airborne Systems Engineering. In: *AIAA Scitech 2020 Forum*. 2020; p. 0896.

- [17] Mahmoodi S, Durak U, Hartmann S, Jafer S. DO-330/ED-215 Overlay to the IEEE Recommended Practice for Distributed Simulation Engineering and Execution Process. *Journal of Aerospace Information Systems*. 2018;15(12):696–705.

SyNumSeS: A Python Package For Numerical Simulation Of Semiconductor Devices

Peter Abele¹, Michael Schäfer²

¹* HTWG Konstanz - University of Applied Sciences, 78462 Konstanz, Alfred-Wachtel-Str. 8, Germany

* pabele@htwg-konstanz.de

² CADwalk GmbH & Co. KG, Joseph-von-Sontheimer-Str. 3, 89604 Allmendingen, Germany

Abstract. SyNumSeS is a Python package for numerical simulation of semiconductor devices. It uses the Scharfetter-Gummel discretization for solving the one dimensional Van Roosbroeck system which describes the free electron and hole transport by the drift-diffusion model. As boundary conditions voltages can be applied to Ohmic contacts. It is suited for the simulation of pn-diodes, MOS-diodes, LEDs (hetero junction), solar cells, and (hetero) bipolar transistors.

Introduction

The development of modern semiconductor devices is accompanied by numerical simulations. This results in a better understanding of the device as physical quantities like electrical fields or current densities can be visualized and related to material parameters or the geometry of the device. This way time consuming experiments can be reduced saving resources, time, and money. In teaching semiconductor devices the visualization of what is happening inside the device is very helpful for the students. So they can get a better link between the learned equations and what do they mean for the electrical properties of the device. Presenting animations showing the output characteristics and for example the hole and electron density in the device for the different bias regions is often better than thousands of words. This was the motivation for programming a simple to use and fast Python package for numerical simulations of semiconductor devices.

This package is used for the lecture 'Electron Devices' to generate diagrams and animations. In the master's course 'Microfabrication Techniques' the students use the package within a lab for the simulations of pn-diodes, LEDs, solar-cells and SiGe hetero bipolar transistors.

1 Solving the basic semiconductor equations

The Van Roosbroeck equations [1] are used as the basic semiconductor equations. For one dimension they are:

$$-\frac{\partial}{\partial x} \left(\epsilon_0 \epsilon_r \frac{\partial}{\partial x} \Psi \right) = q_e \cdot (C + p - n), \quad (1a)$$

$$\frac{\partial}{\partial t} n = \frac{1}{q_e} \cdot \frac{\partial}{\partial x} j_n + (G - R), \quad (1b)$$

$$\frac{\partial}{\partial t} p = -\frac{1}{q_e} \cdot \frac{\partial}{\partial x} j_p + (G - R). \quad (1c)$$

These are three nonlinear ordinary differential equations for three unknowns, the potential Ψ , the free electron density n , and the free hole density p . (C is the doping concentration, G and R are the generation and recombination rates, respectively, ϵ_0 is the dielectric constant, and ϵ_r is the relative permittivity.) These equations can be solved by using partial differential equation (PDE) solvers like FiPy[2]. As for semiconductor devices high electrical fields and extreme concentration gradients for electrons and holes exist especially at pn- or hetero-junctions using a general PDE solver results in significant deviation from the exact solution. This can be partly compensated by reducing the grid size but at the expense of simulation time and even convergence issues.

By using the Scharfetter Gummel scheme [3][4] for the electron and hole current densities this issue can be relaxed.

Additionally in the equations (1) the unknowns n and p are replaced by the quasi Fermi levels for electrons Φ_n and holes Φ_p , respectively using the flowing relations:

$$n = N_C \cdot \exp\left(\frac{X - \Phi_n + \Psi}{U_t}\right) \quad (2)$$

and

$$p = N_V \cdot \exp\left(\frac{-X - E_g + \Phi_p - \Psi}{U_t}\right). \quad (3)$$

With X^1 being the electron affinity, E_g the band gap, and $U_t = \frac{k_B T}{q_e}$ (k_B Boltzmann constant, T absolute temperature, and q_e elementary charge). N_C and N_V are the effective density of states for the conduction band and the valence band, respectively.

For solving this PDEs the finite volume method for one dimension is used. The electron and hole current densities between two adjacent cells are expressed by:

$$j_{n;k,l} = -\frac{q_e U_t}{\Delta x} \cdot \frac{\mu_{n,k} + \mu_{n,l}}{2} \cdot \frac{N_{C,k} + N_{C,l}}{2} \cdot \exp\left(\frac{X_k + X_l}{2}\right) \cdot \left[B\left(-\frac{\Psi_l - \Psi_k}{U_t}\right) \cdot \exp\left(\frac{\Psi_k - \Phi_{n,k}}{U_t}\right) - B\left(\frac{\Psi_l - \Psi_k}{U_t}\right) \cdot \exp\left(\frac{\Psi_l - \Phi_{n,l}}{U_t}\right) \right] \quad (4)$$

and

$$j_{p;k,l} = -\frac{q_e U_t}{\Delta x} \cdot \frac{\mu_{p,k} + \mu_{p,l}}{2} \cdot \frac{N_{V,k} + N_{V,l}}{2} \cdot \exp\left(\frac{X_k + E_{g,k} + X_l + E_{g,l}}{2}\right) \cdot \left[B\left(-\frac{\Psi_l - \Psi_k}{U_t}\right) \cdot \exp\left(\frac{\Phi_{p,k} - \Psi_k}{U_t}\right) - B\left(\frac{\Psi_l - \Psi_k}{U_t}\right) \cdot \exp\left(\frac{\Phi_{p,l} - \Psi_l}{U_t}\right) \right] \quad (5)$$

With k and l being indices of adjacent cells, μ_n and μ_p are the mobilities of electrons and holes, respectively, and Δx is the cell distance. $B(x)$ is the Bernoulli function:

$$B(x) = \frac{x}{\exp(x) - 1}. \quad (6)$$

Compared to [4] the equations (4) and (5) are slightly modified by the terms containing the electron affinity X to account for hetero junctions.

Care must be taken for the Bernoulli function in the case of low values for the argument as this results in a

low value divided by an other low value. Therefore the Bernoulli function is approximated by a polynomial of degree 1 for low values of the argument:

$$B(x) = 1 - \frac{x}{2}. \quad (7)$$

2 Boundary condition

On the very left and right of the one dimensional device boundary conditions for Ohmic contacts are set and different voltages can be applied to both sides. This is done by adding one additional cell on the left and right with the quasi Fermi levels and potential set to [5]:

$$\Phi_n = V_A \quad (8a)$$

$$\Phi_p = V_A \quad (8b)$$

$$\Psi = -X - \frac{E_g + U_t \log\left(\frac{N_C}{N_V}\right)}{2} + U_t \operatorname{arcsinh}\left(\frac{C}{2n_i}\right) + V_A \quad (8c)$$

With V_A the applied voltage and n_i the intrinsic carrier density. For the simulation of bipolar transistors an additional contact between the outer contacts is needed. For this contact dependent on the doping, p- or n-type, only the potential Ψ and one of the quasi Fermi levels Φ_p or Φ_n are set, respectively.

3 Solving the semiconductor equations

The discretization of the equations (1) using the equations (2)-(5) results in a nonlinear equation system for the unknowns Ψ , Φ_n , and Φ_p for every cell. To solve this nonlinear equation system Newton's method is used. For building the Jacobian matrix the equations are symbolically derived using the Python package sympy. The code for solving these equations is automatically generated by a Python script.

¹ For ease of use the units of X , E_g , Φ_n , and Φ_p are V and not eV!

4 Simulation results

To demonstrate the wide use of this package simulation results for three different semiconductor devices are presented.

4.1 Simulation results of an InGaN LED

For the simulation of an LED² the InGaN material system is used. The well, where the light is generated, consists of $\text{In}_{0.3}\text{Ga}_{0.7}\text{N}$ with a band gap of about $E_g = 2.0\text{eV}$ and has a thickness of 100nm . The surrounding material is GaN with $E_g = 3.4\text{eV}$. The $\text{In}_{0.3}\text{Ga}_{0.7}\text{N}$ well is not doped while the surrounding GaN to the left is p-doped and to the right is n-doped both with a concentration of $N_A = N_D = 2 \cdot 10^{22}\text{m}^{-3}$.

Fig. 1 shows the resulting band diagram with no externally applied voltage.

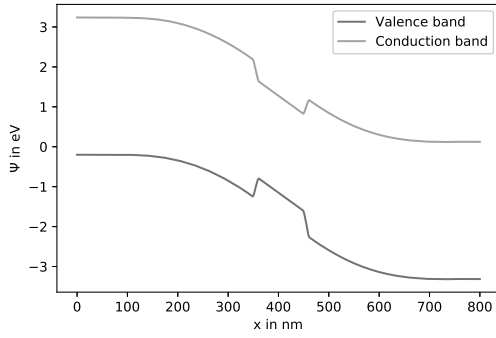


Figure 1: Band diagram of InGaN-LED with no externally applied voltage.

In the center the $\text{In}_{0.3}\text{Ga}_{0.7}\text{N}$ well can be clearly seen and the band bending respectively build in voltage Φ_{bi} due to the doping concentrations. Simulating the output characteristics shows a diode threshold voltage of about $V_{th} = 2.8\text{V}$, see Fig. 2.

The nearly linear increase of the current density beyond V_{th} can be attributed to the low doping level of the GaN resulting in a relatively high Ohmic resistance.

The recombination rate is considered by:

$$R = C_{rad} \cdot (np - n_i^2). \quad (9)$$

As the free electrons and holes accumulate at the well we get in this region the highest recombination rate, see Fig. 3.

²Light emitting diode

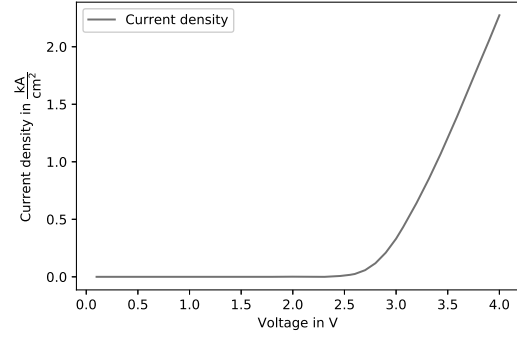


Figure 2: LED characteristic

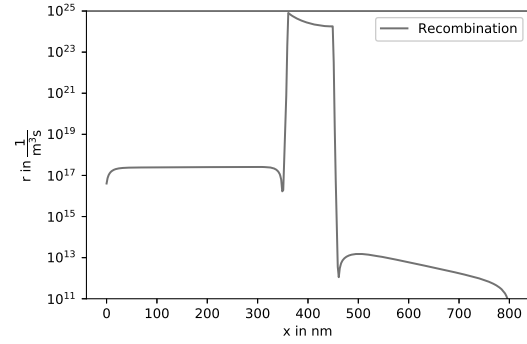


Figure 3: Recombination rate along the LED

4.2 Simulation results of solar cells

In the following silicon solar cells are simulated. The thickness of the silicon is $d = 1\mu\text{m}$. The first 40nm from the surface are p-doped with $N_A = 1.0 \cdot 10^{24}\text{m}^{-3}$. The rest of the silicon is n-doped with different concentrations. The intensity of the incident light drops exponentially while penetrating through the silicon and generate free electrons and holes.

In the following the influence of the different n-doping concentrations, $N_D = 1.25 \cdot 10^{21}\text{m}^{-3}$, $N_D = 1.0 \cdot 10^{22}\text{m}^{-3}$, and $N_D = 8.0 \cdot 10^{22}\text{m}^{-3}$ is shown. The intensity of the incident light is kept constant.

The doping level has two effects, first on the build in voltage Φ_{bi} , which increases with increasing doping level, and second the extension of the space charge region which decreases with increasing doping level. A higher build in voltage results in a higher idle voltage, this can be seen in Fig. 4. But on the other side with a reduction of the extension of the space charge region

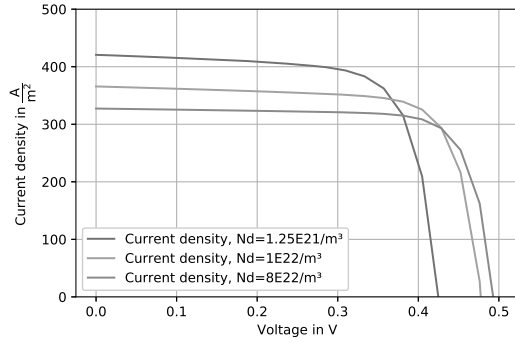


Figure 4: Solar cell output characteristic for different n-doping concentrations

the short circuit current drops.

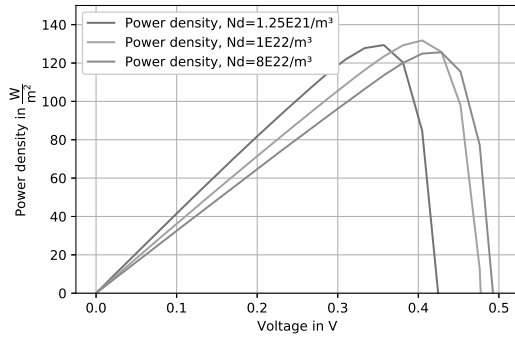


Figure 5: Output power density of differently doped solar cells

For optimizing the doping concentration in Fig. 5 the power density dependent on the applied voltage is shown. The maximum power point is for the doping levels of $N_D = 1.25 \cdot 10^{21} \text{ m}^{-3}$ and $N_D = 1.0 \cdot 10^{22} \text{ m}^{-3}$ about the same. For the higher doping level it is about 5% lower.

4.3 Simulation results of a Si bipolar transistor

The following shows simulation results of a npn bipolar transistor. The device has an extension of $1.2 \mu\text{m}$. The first 400nm build the emitter which is n-doped with $N_{D,E} = 1.2 \cdot 10^{24} \text{ m}^{-3}$. The next 400nm build the base which is p-doped with $N_{A,B} = 4.0 \cdot 10^{22} \text{ m}^{-3}$. The last 400nm build the collector which is n-doped with

$$N_{D,C} = 2.0 \cdot 10^{22} \text{ m}^{-3}.$$

For this simulation the base contact is located at $x = 650\text{nm}$. As SyNumSeS can only solve devices for one dimension, the boundary condition at the base contact (p-doped) only fixes the potential Ψ and the quasi Fermi level for the holes Φ_p . The quasi Fermi level for the electrons Φ_n at this point is calculated by the nonlinear equation system. Thereby the electrons can pass the base contact.

In a first step the base emitter voltage is ramped from $V_{BE} = 0\text{V}$ to $V_{BE} = 0.6\text{V}$ in steps of 0.1V . Then the collector emitter voltage is ramped to $V_{CE} = 3.0\text{V}$ in steps of 50mV . This results in the output characteristics and is show in Fig. 6.

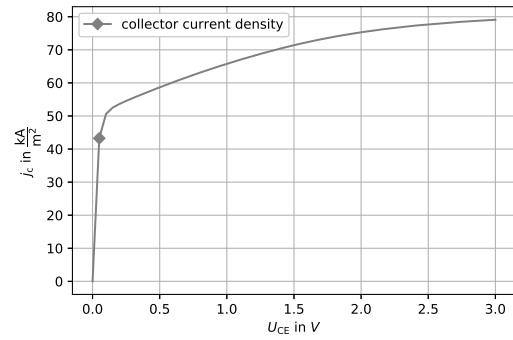


Figure 6: Output characteristics of a Si bipolar transistor for $V_{BE} = 0.6\text{V}$

The simulated output characteristics shows the saturation region up to about $V_{CE} = 100\text{mV}$. The following slope can be attributed to the Early effect.

To get a better understanding for the saturation region, we can have a look at the electron and hole current density across the device at $V_{CE} = 50\text{mV}$ and $V_{BE} = 0.6\text{V}$, see Fig. 7.

It can be clearly seen that electrons are not only injected from the emitter into the base, but also from the collector into the base ($400\text{nm} < x_{\text{Base}} < 800\text{nm}$). So the gradient of the electron density is low, resulting in a low collector current caused by diffusion.

5 Summary

The Python package SyNumSeS has proven its capability for the simulation of various semiconductor devices. As these devices are also discussed in bachelor

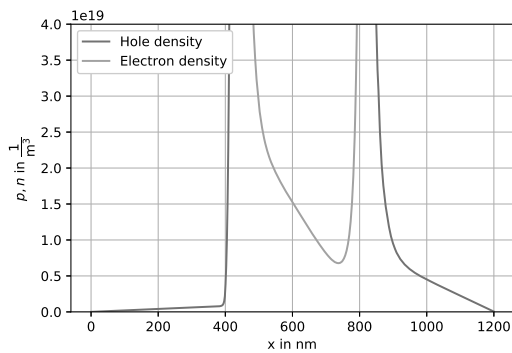


Figure 7: Electron and hole density across the bipolar transistor at saturation for $V_{CE} = 50\text{mV}$ and $V_{BE} = 0.6\text{V}$

and master courses the package is well suited to support the lecture by visualizing physical quantities in the device for different bias regions. Due to its fast simulation times it is also appropriate for lab courses where the students can investigate the influence of process and material parameters on the performance of a device.

This Python package can be installed by:

```
pip install synumses-pkg-pabele
```

and cloned from

<https://github.com/pabele/synumses>

with several examples using the JupyterLab notebook.

References

- [1] Van Roosbroeck W. Theory of the Flow of Electrons and Holes in Germanium and Other Semiconductors. *Bell System Technical Journal*. 1950;29(4):560–607.
- [2] Guyer JE, Wheeler D, Warren JA. FiPy: Partial Differential Equations with Python. *Computing in Science & Engineering*. 2009;11(3):6–15.
- [3] Scharfetter DL, Gummel HK. Large-signal analysis of a silicon Read diode oscillator. *IEEE Transactions on Electron Devices*. 1969;16(1):64–77.
- [4] Farrell P, Rotundo N, Doan DH, Kantner M, Fuhrmann J, Koprucki T. Drift-Diffusion Models. In: *Handbook of Optoelectronic Device Modeling and Simulation*, edited by Piprek J, pp. 733–772. London, New York: CRC Press. 2017;.
- [5] Selberherr S. *Analysis and Simulation of Semiconductor Devices*. Wien, New York: Springer. 1984.

Orthotropic non-hysteretic permeability model as a lookup-table for FEA

Robert Courant^{1*}, Jürgen Maas¹

¹ Mechatronic Systems Laboratory, Technische Universität Berlin, 10623 Berlin, Germany

*robert.courant@emk.tu-berlin.de

Abstract. Within this paper, we present an appropriate approach to prepare the data of orthotropic nonlinear soft magnetic materials for use in finite element analyses. Based on the co-energy density, an elliptic model for the interpolation between the principal directions is derived and extended to laminated materials. Instead of adapting the FE-code to incorporate the fundamental equations, a workflow is proposed to generate a lookup-table of the anisotropic nonlinear permeability which can be easily integrated in FE-programs like COMSOL Multiphysics. The modelled behaviour is shown for typical material classes and the lamination model is numerically validated.

Introduction

Soft magnetic materials are essential for almost all magnetic circuits. They play an important role in electromotors and transformers to guide the flux to guarantee an efficient operation[1],[2]. For the dimensioning of those circuits, accurate models of their magnetic behaviour are necessary. The most important characteristic is the relation between magnetic field strength \mathbf{H} and magnetic flux density \mathbf{B} . All ferromagnetic materials show a saturation at high fields, therefore a nonlinear approach has to be chosen. As the nonlinear problems are almost impossible to solve analytically for complex geometries, magnetic problems are typically computed numerically.

High performance soft magnetic materials often exhibit an anisotropic behaviour. Characteristic examples are different grain oriented (GO) electrical steels that show a magnetic easy axis in rolling direction and transverse magnetic hard axes. Consequently, magnetically anisotropic materials are typically characterised by uniaxial nonlinear curves $B_{u,i}(H_{u,i})$ in three principal directions. This experimental characterisation of anisotropic materials requires sophisticated measurements[3],[4]. Because the flux components influence each other (even

for isotropic materials), for vectors between the principal directions, a suitable interpolation has to be found. An extension of the topic are laminated materials, where non-ferromagnetic layers inhibit the perpendicular flux more than in plane.

This paper chooses an elliptical approach based on the co-energy density, which is well known concept[5],[6]. Key elements are based on the implementation of [7]. It has to be noted that hysteresis is not considered here. For magnetic materials, hysteresis is typically modelled with a JILES-ATHERTON model, other models like PREISACH or PRANDTL-ISHLINSKI are also applicable.

Another application of anisotropic magnetic properties, that attracted the author to the topic in the first place, are magnetic shape memory alloys, studied for their shape memory and magnetocaloric effect. These are indeed described by BH-curves in easy direction and in the perpendicular hard directions as shown in Figure 1. Therefore, they are appropriate benchmarking examples for our model.

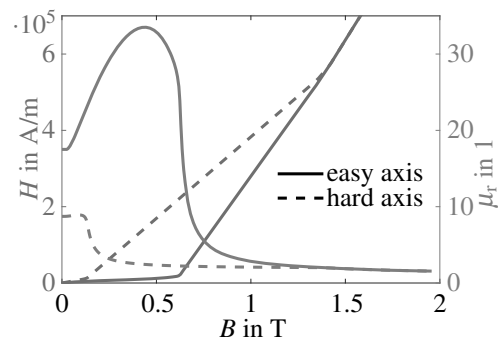


Figure 1: HB- and $\mu_r(B)$ -curves for easy and hard axis of examined MSMA from [8].

Section 1 introduces the physical background and motivates the elliptical model. This approach is generalized for nonlinear orthotropic materials in section 2. Section 3 extends the model to laminated materials. Be-

cause magnetic anisotropy can be only incorporated as a relative permeability in the used FE-software COMSOL Multiphysics, the model is prepared as a look-up table of the nonlinear permeability dependent on the flux density \mathbf{B} . In section 4, our model is analysed for different example materials. The laminate is numerically validated in section 5.

1 Constitutive Equations

Electromagnetic problems are described by MAXWELL's equations. There, two quantities each are defined for the electric and magnetic domain, one for the excitation and one for the effect. In the magnetic domain, these are the magnetic field strength \mathbf{H} and the magnetic flux density \mathbf{B} . These two vector fields are coupled by generally nonlinear material properties

$$\mathbf{B} = \mathbf{B}(\mathbf{H}). \quad (1)$$

The co-energy density of the magnetic field can be derived by

$$w'(\mathbf{B}) = \int_0^{\mathbf{B}} \mathbf{H} \cdot d\tilde{\mathbf{B}}. \quad (2)$$

In the easiest case of vacuum this is

$$w'(\mathbf{B}) = \frac{\mathbf{B}^2}{2\mu_0} \quad (3)$$

which leads to the linear coupling of \mathbf{B} and \mathbf{H}

$$\mathbf{B} = \mu_0 \mathbf{H}. \quad (4)$$

For general linear materials, this is extended using Einstein notation to

$$B_j = \mu_0 \mu_{r,ij} H_i \quad (5)$$

with the relative permeability matrix $\mu_{r,ij}$. In most cases, the relative permeability is assumed orthotropic and the matrix is thus

$$\frac{1}{\mu_{r,ij}} = \begin{pmatrix} \frac{1}{\mu_{r,x}} & 0 & 0 \\ 0 & \frac{1}{\mu_{r,y}} & 0 \\ 0 & 0 & \frac{1}{\mu_{r,z}} \end{pmatrix}. \quad (6)$$

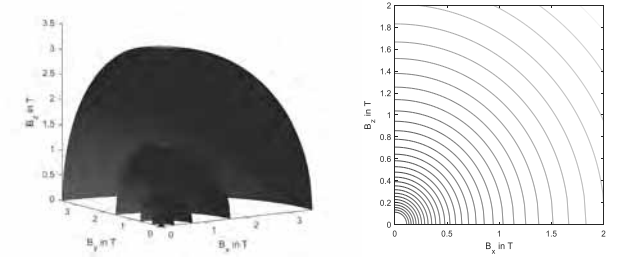


Figure 2: Contour lines of the magnetic co-energy for a linear isotropic material with $\mu_{r,iso} = 5000$.

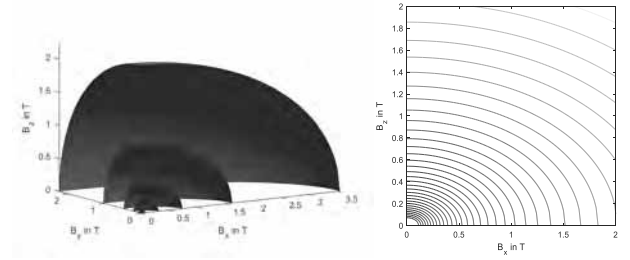


Figure 3: Contour lines of the magnetic co-energy for a linear anisotropic material.

The co-energy density for this case is

$$\begin{aligned} w'(\mathbf{B}) &= \int_0^{\mathbf{B}} \frac{1}{\mu_0 \mu_{r,ij}} \tilde{B}_j \cdot d\tilde{B}_i \\ &= \frac{B_x^2}{2\mu_0 \mu_{r,x}} + \frac{B_y^2}{2\mu_0 \mu_{r,y}} + \frac{B_z^2}{2\mu_0 \mu_{r,z}}. \end{aligned} \quad (7)$$

The contour surfaces $w' = \text{const.}$ of this are concentric ellipsoids. This is why an elliptic model is used in section 4 to model anisotropic nonlinear functions. Figure 2 depicts the contour surfaces in 3D or contour lines in 2D for a linear isotropic material with $\mu_{r,iso} = 5000$. The contour surfaces are concentric spheres. Figure 3 shows the expected ellipsoids for a linear anisotropic material with $\mu_{r,x} = 5000$ and $\mu_{r,y} = \mu_{r,z} = 2000$.

Nonlinear isotropic materials have equally oriented field and flux, where the absolute value is described by a nonlinear relation as

$$\mathbf{B} = B_{iso}(|\mathbf{H}|) \frac{\mathbf{H}}{|\mathbf{H}|}. \quad (8)$$

The use of the absolute value produces a spherical interpolation between the directions. Although the shown relation $\mathbf{B}(\mathbf{H})$ is the more comprehensible, for numerical

validations the opposite formulation $\mathbf{H}(\mathbf{B})$ is preferable, because in most FE-programs, the problem is solved using the auxiliary vector potential \mathbf{A} that is linked to the flux density by $\mathbf{B} = \text{rot}(\mathbf{A})$. Accordingly, in section 3 we will setup the lookup-table for the nonlinear function of μ_r dependent on \mathbf{B} in the form of

$$(B_x|B_y|B_z|\mu_{r,x}|\mu_{r,y}|\mu_{r,z}). \quad (9)$$

2 Elliptic Interpolation

The interpolation for nonlinear orthotropic materials has to fulfil three conditions. Firstly, it has to be a convex function, which translates to convex contour lines in 2D or surfaces in 3D. Secondly, the interpolation has to match the uniaxial nonlinear curves it is fitted to. Finally, the approach has to match the elliptical model of the co-energy density as shown in (7) for linear orthotropic magnetic materials. An obvious choice, that meets the requirements, is again an elliptical model of the co-energy density $w'(\mathbf{B})$, where the half-axes in all directions of the contour surface for each co-energy density w' are determined by the corresponding flux density $B_{u,i}(w')$ of the uniaxial measurements.

These measurements are typically provided as a table $(H_{u,i}|B_{u,i})$. The co-energy densities can be numerically integrated according to (2). From the extended table $(H_{u,i}|B_{u,i}|w')$ a function $B_{u,i}(w')$ can be interpolated. For each w' , the contour surface is described by

$$\left(\frac{B_x}{B_{u,x}(w')}\right)^2 + \left(\frac{B_y}{B_{u,y}(w')}\right)^2 + \left(\frac{B_z}{B_{u,z}(w')}\right)^2 = 1. \quad (10)$$

Figure 4 illustrates the two-dimensional case with a contour line. The interpolation scheme is to generate preferably equally spaced sample points of the ellipsoids for each co-energy density $(B_x|B_y|B_z|w')$. These values can be interpolated to a function $w'(\mathbf{B}_x, \mathbf{B}_y, \mathbf{B}_z)$.

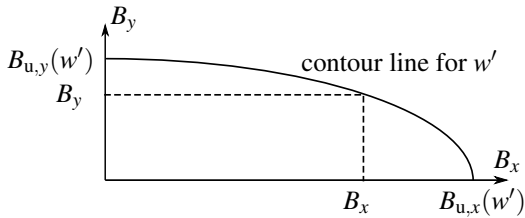


Figure 4: 2D representation of the elliptic interpolation.

3 Lamination

Laminated magnetic materials could be modelled with their full geometry as shown in section 5. Due to the thin layers, especially of the non-magnetic insulation, an otherwise unnecessarily fine mesh is required which increases the computational burden. Therefore, for most applications a homogenised material model suffices. The insulation material 2 is non-ferromagnetic $\mu_r = 1$. The stacking factor f is the volume ration of the ferromagnetic material 1 to the overall volume. The sheets are thin, a change in the field from one sheet to the next can be neglected. Without loss of generality, the sheets are assumed to be in the xy-plane and stacked in z direction. At interfaces, the normal component of the flux density is continuous, therefore the homogenized flux density is

$$B_z = B_{1,z} = B_{2,z}. \quad (11)$$

The homogenized flux density in the xy-plane is constructed by a mixing law

$$\begin{aligned} B_x &= fB_{1,x} + (1-f)B_{2,x} \\ B_y &= fB_{1,y} + (1-f)B_{2,y}. \end{aligned} \quad (12)$$

The homogenized co-energy density w'_{lam} can be calculated as a weighted sum of that of the ferromagnetic material $w'_1(B_{1,x}, B_{1,y}, B_{1,z})$ from the interpolation and that of vacuum from (3) and is dependent on the homogenized flux density and that in the insulation

$$\begin{aligned} w'_{\text{lam}}(B_x, B_y, B_z, B_{2,x}, B_{2,y}) \\ = fw'_1\left(\frac{1}{f}(B_x - (1-f)B_{2,x}), \frac{1}{f}(B_y - (1-f)B_{2,y}), B_z\right) \\ + \frac{1-f}{2\mu_0}(B_{2,x}^2 + B_{2,y}^2 + B_z^2). \end{aligned} \quad (13)$$

For a strong ferromagnetic material 1, $B_{2,x}$ and $B_{2,y}$ can be neglected and the approximated co-energy density is

$$w'_{\text{lam}}(\mathbf{B}) = fw'_1\left(\frac{B_x}{f}, \frac{B_y}{f}, B_z\right) + \frac{1-f}{2\mu_0}B_z^2. \quad (14)$$

Relative permeability. The relative permeabilities for the look-up table (9) could be calculated using (5) with the field strength \mathbf{H} derived according to (2) as

$$\mathbf{H}(\mathbf{B}) = \frac{\partial w'_{\text{lam}}(\mathbf{B})}{\partial \mathbf{B}}. \quad (15)$$

While this is physically correct, the numerical derivate on the interpolated function $w'_{\text{lam}}(\mathbf{B})$ is not very accurate for a realistic sampling grid. Instead, the permeability can be computed from the uniaxial measurements ($H_{u,i}|B_{u,i}$). Using the analytical derivate of (14), the uniaxial field strengths of the complete laminated material can be calculated as

$$\begin{aligned} H_{\text{lam},u,x}(B_{u,x}) &= H_{u,x}\left(\frac{B_{u,x}}{f}\right) \\ H_{\text{lam},u,y}(B_{u,y}) &= H_{u,y}\left(\frac{B_{u,y}}{f}\right) \\ H_{\text{lam},u,z}(B_{u,z}) &= H_{u,z}(B_{u,z}) + \frac{1-f}{\mu_0} B_{u,z}. \end{aligned} \quad (16)$$

Now, (5) can be solved for $\mu_{r,\text{lam},i}(B_{u,z})$. For the uniaxial values, the ferroelectric material's contribution to the co-energy density

$$w'_{\text{an}} = w'_1\left(\frac{B_x}{f}, \frac{B_y}{f}, B_z\right) \quad (17)$$

can be accurately integrated according to (2). While the overall behaviour of the laminate is dependent on the complete co-energy density w'_{lam} , the relative permeabilities of the ferroelectric material are only dependent on the new co-energy density w'_{an} . Therefore, the permeabilities of the ferroelectric material and consequently of the laminate can be derived from the uniaxial measurements. Consequently, the generation of the look-up table (9) consists of two steps: First, $w'_{\text{an}}(B_x, B_y, B_z)$ is calculated on a grid ($B_x|B_y|B_z$), then the permeabilities ($\mu_{r,\text{lam},x}|\mu_{r,\text{lam},y}|\mu_{r,\text{lam},z}$) are taken from the uniaxial measurement with the corresponding co-energy density w'_{an} . The presented approach yields a far smoother permeability than the direct derivation according to (15), especially at low flux densities.

4 Application

Applying our method to the example depicted in Figure 3 shows no difference. Therefore, the third condition is fulfilled. The second condition can be tested by comparing μ_r along an axis from the elliptic model with the uniaxial input data. The interpolation shall be applied to a representative GO electrical steel of grade M-6. The BH-curve is taken from the COMSOL material library, the results are shown in Figure 5. The rolling direction has a much higher initial permeability, proportional to the derivate of the curve, than the transverse

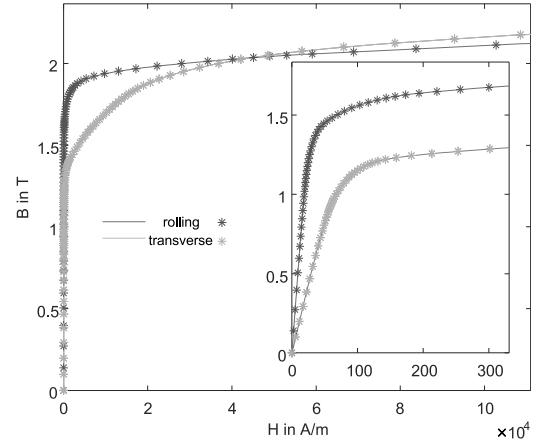


Figure 5: BH-curve of the investigated grain oriented steel M-6 in both principal directions, detail on the right.

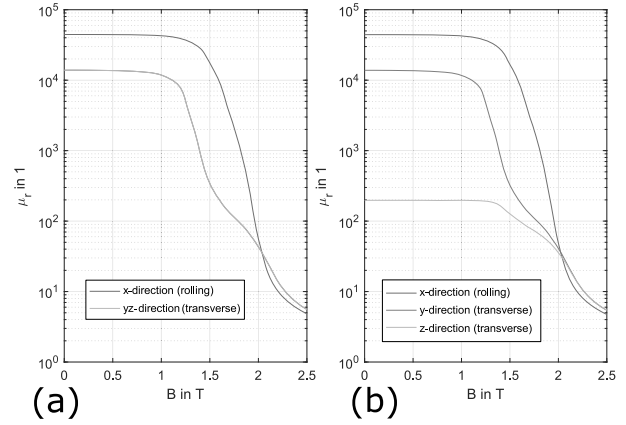


Figure 6: $\mu_{r,i}(B_{r,i})$ in all directions, (a) of the investigated GO steel M-6 and (b) of the laminate with stacking factor $f = 0.995$.

direction. At very high fields, the transverse direction gains a slight advantage. This can also be seen in Figure 6 (a) of the elliptic model for this material with the rolling direction in x and the transverse curves used for y and z . The curves of the uniaxial measurements perfectly match the model, therefore the second condition is fulfilled.

The contour surfaces or contour lines in a 2D-section for the elliptical model of this material can be seen in Figure 7. At lower fields, the difference between the axes is higher, almost reaching a factor of two, while the results approximately match at higher fields in accordance with the input curves. The density of the contour lines in the 2D-section is correlated with

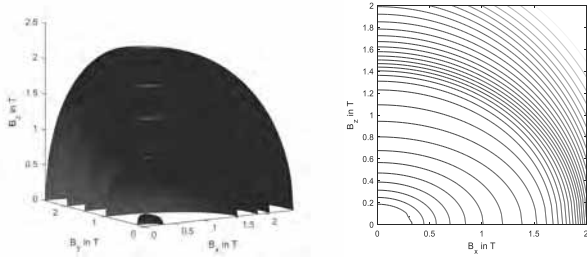


Figure 7: Contour lines of the magnetic co-energy density for pure grain oriented steel without lamination.

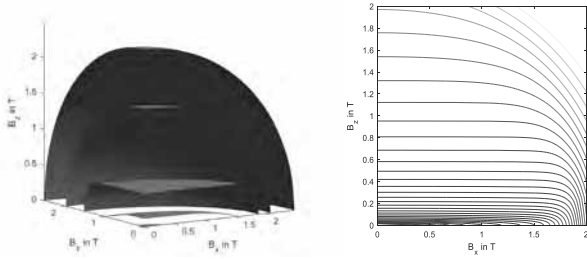


Figure 8: Contour lines of the magnetic co-energy density for laminated grain oriented steel with $f = 0.995$.

the permeability. At low fields, the permeability is high and the energy doesn't grow that fast, after the saturation, more energy is needed to still increase the field. When the axes roughly match for high fields, the contour lines are almost concentric circles.

In Figure 8 and Figure 9, laminates of the same grain oriented steel with different stacking factors are shown. In Figure 8, the stacking factor is $f = 0.995$, meaning that only 0.5% are non-magnetic insulation. Nevertheless, for low co-energy densities the flux in z -direction changed drastically. The former ellipses in 2D

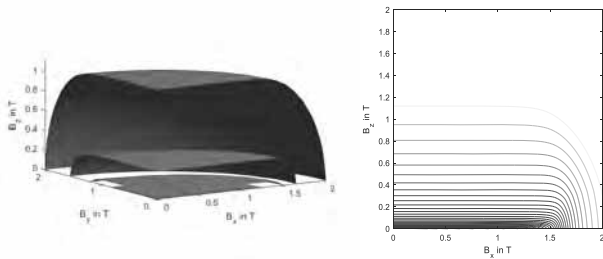


Figure 9: Contour lines of the magnetic co-energy density for laminated grain oriented steel with $f = 0.8$.

and ellipsoids in 3D are flattened with an almost parallel top. The corresponding relative permeabilities can be seen in Figure 6 (b). Because of the almost identical cross section compared to the full GO material, the in-plane curves x and y are also almost identical between (a) and (b). In contrast to that relative permeability in stacking direction z is greatly reduced. With increasing saturation, the effect of the gap becomes less prominent, until both transverse directions are almost identical again. The contour surfaces or contour lines of the ferroelectric material's contribution to the co-energy density w'_{an} according to (17) are almost indistinguishable from the co-energy density of the pure GO steel without lamination in Figure 7 and are therefore not separately shown. This is again because of the almost identical cross section.

The effect extends to higher co-energy densities with the lower stacking factor $f = 0.8$ in Figure 9. The field in z is dominated by the insulation even for higher fields where the ferromagnetic materials begins to reach the saturation. With the reduced cross section area of the ferromagnetic material, the flux in x and y is also slightly reduced and the saturation is reached a bit earlier.

Another example to test our model are magnetic shape memory alloys (MSMA). Figure 1 shows the characteristic HB-curves for the easy and hard axis. The results of the elliptic model are depicted in Figure 10. While the initial permeability of the GO steel was around 40000 in rolling direction, the MSMA only reaches 30. Therefore, higher co-energy densities are evaluated to cover the same flux density range. The MSMA's anisotropy is larger than that of the GO steel. The change in permeability isn't as high, consequently the contour lines in 2D are more equally spaced.

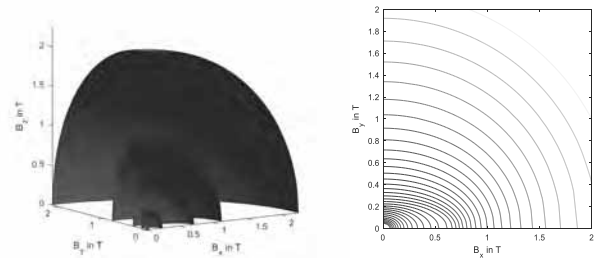


Figure 10: Contour lines of the magnetic co-energy density for MSMA with easy axis in x .

5 Numerical validation

To validate the laminated model, a benchmark FE-simulation is conducted in COMSOL, comparing the geometrically modelled laminate with an averaged continuum using the generated look-up table (9) at different angles with $\varphi = 0$ denoting the flux in stacking direction z and $\varphi = 90$ for the flux in rolling direction x . For a decent accuracy, a table with 200 flux densities in each direction is used. As described in the introduction, magnetically anisotropic materials have to be implemented as relative permeabilities dependent on the flux density \mathbf{B} . In contrast to that, the materials from the included material library are implemented as isotropic BH-curves as shown in Figure 5. To exclude errors in the developed anisotropic model, the comparison is therefore conducted with the isotropic material of GO M-6 in rolling direction from the same diagram. For the laminate, a stacking factor $f = 0.95$ is investigated.

Additionally the relative permeability of the laminate rotated by φ is calculated analytically. The co-energy density w'_{an} in the ferromagnetic material is

$$w'_{\text{an}}(\mathbf{B}, \varphi) = w'_{\text{an}}(B \sin(\varphi), 0, B \cos(\varphi)). \quad (18)$$

Using the look-up table, the relative permeabilities in material coordinates $\mu_{r,i,\text{mat}}$ can be determined. Using those, the magnetic fields in material coordinates are

$$\begin{aligned} H_{x,\text{mat}}(\mathbf{B}, \varphi) &= \frac{B \sin(\varphi)}{\mu_{r,x,\text{mat}} \cdot \mu_0} \\ H_{z,\text{mat}}(\mathbf{B}, \varphi) &= \frac{B \cos(\varphi)}{\mu_{r,z,\text{mat}} \cdot \mu_0}. \end{aligned} \quad (19)$$

The analytical field in flux direction and the relative permeability are then

$$\begin{aligned} H_{\text{ana}}(\mathbf{B}, \varphi) &= \sin(\varphi) H_{x,\text{mat}} + \cos(\varphi) H_{z,\text{mat}} \\ \mu_{r,\text{ana}}(\mathbf{B}, \varphi) &= \frac{B}{H_{\text{ana}} \cdot \mu_0}. \end{aligned} \quad (20)$$

The model setup in COMSOL is shown in Figure 11. It consists of four tests, evaluated in the test domain between the orange lines and connected parallelly to the same excitation. On the left, a material using the look-up-table of the laminated material is defined (magenta). To measure the material response at different angles, the material coordinates are rotated accordingly. This test of the averaged lamination model is designated "avg". The other three tests are different implementations of

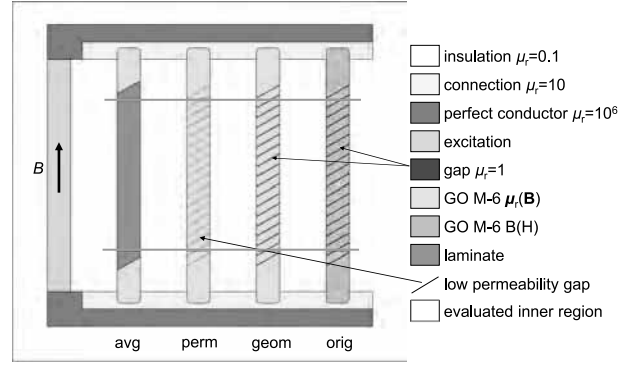


Figure 11: Model setup for rotation angle $\varphi = 30$ with used materials and test designation at the bottom.

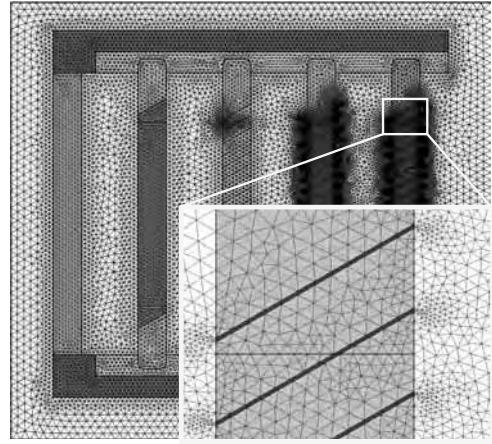


Figure 12: Used mesh, detail of the fully modelled laminate with two elements over the gap.

the complete stacked geometry, which is again rotated for different cases.

The test designated "perm" uses a so-called "low permeability gap" between the layers of GO M-6. This special transition condition is used to model a thin gap of non-ferromagnetic material on a boundary instead of in a domain. An advantage is the easier mesh without the thin domains, the internal behaviour is not automatically calculated though. While the magnetic flux is computed correctly, the field strength and energy in the gap have to be considered manually to calculate the average permeability of the laminate.

The last two tests designated "geom" and "orig" use the fully modelled gaps. This can be seen in the detail of Figure 12, where the thin blue gap with $\mu_r = 1$ is shown. The gaps are identical for both tests, but the isotropic GO material is modelled differently. While for "geom"

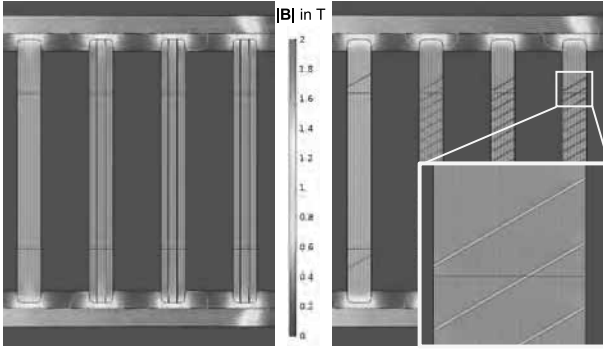


Figure 13: Resulting fields for an excitation of 5T, orientation of the laminate left $\varphi = 90$, right $\varphi = 30$, fluxlines in light grey.

the look-up table is used, "orig" is modelled directly using the aforementioned BH-curve from Figure 5.

The excitation has to be connected to the angled tests in a numerically stable way for a wide flux range. This is done in three steps. First, a vertical remanent flux is defined in the green excitation domain, this flux is distributed through the red almost perfect conductor with $\mu_r = 10^6$. To avoid steep gradients at the interface to the laminated material especially at $\varphi = 90$ (see Figure 13 on the left), a connection layer with a lower permeability of $\mu_r = 10$ is defined and the corners are rounded. After a short layer of the non-laminated GO material, the test materials start outside the evaluated inner region. The necessary air region around the flux guiding geometries has a reduced permeability of $\mu_r = 0.1$. While this magnetic insulation is unphysical, it helps to concentrate the flux inside the test domains even at high saturation. Without this addition, the flux inside the test domains would be less uniformly vertical. We can therefore be even more accurate at high flux densities than physical experiments with the same geometry could be. The mesh can be seen in Figure 12. While a sufficiently fine mesh is defined for most domains, special care is required at the thin gaps in "geom" and "orig". These are done with a mapped mesh with two elements over the gap as shown in the detail view.

The local flux densities for two load cases can be seen in Figure 13. Both show an excitation of 5T which results in approximately 1.8T in the test domains. Therefore, the GO steel is already in the beginning saturation according to Figure 6. While the overall flux is oriented vertically in the test domains in all cases, the detail at $\varphi = 30$ shows a zig-zag pattern of the (grey)

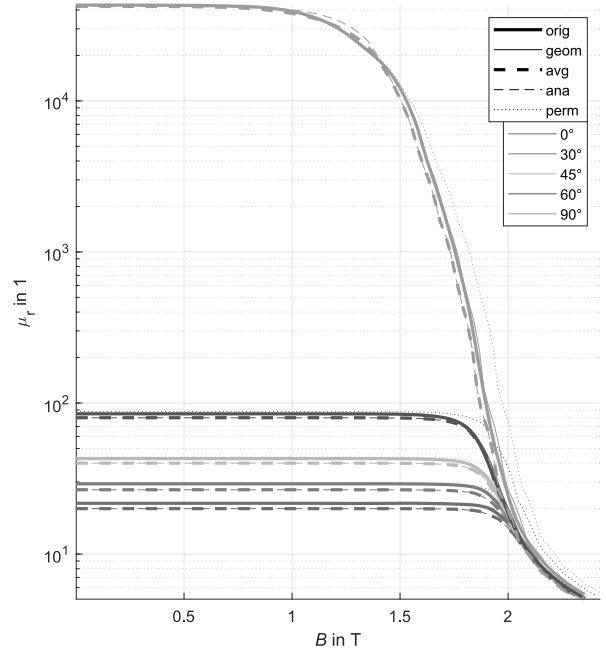


Figure 14: Resulting relative permeability $\mu_r(B)$ for the specified tests (line styles) and angles (line colours).

flux lines between the GO regions and the gaps. Therefore, the bridged length of the gap is reduced and accordingly the magnetic resistance. A much thinner, almost one-dimensional geometry, would reduce this effect. In turn a sleek test region would suffer more flux leakage through the adjacent air regions.

The reduced magnetic resistance of the geometrically modelled tests "geom" and "orig" for diagonal layers compared to the averaged model can be also seen in the resulting relative permeabilities in Figure 14. There, the angles are colour coded and the four test scenarios and the analytic calculation according to (20) are denoted with different line styles. The two geometrically modelled tests "geom" and "orig" are shown with full lines, with a thicker line for "orig", the implementation using the original material model. Apart from some ripples in the saturation region a complete fit can be observed. Those ripples are caused by the interpolation on the look-up table and dependent on its resolution.

The simulated averaged model is marked with a thick dashed line. The thin dashed line denotes the test "ana". This is the analytical calculation according to (20) applied to the average flux density in the test domain. Apart from a slight deviation at the beginning saturation at $\varphi = 90$, the analytical calculation "ana"

and the implemented model "avg" are a complete match. The ripples in both curves are again caused by the resolution of the look-up table. The dotted line denoting the test "perm" with the low permeability gap overestimates the permeability at higher flux densities in rolling direction (higher angles towards $\varphi = 90$), because the reduced cross section of the ferromagnetic material cannot be taken into account in this setup. This deviation would increase for higher stacking factors.

The overall performance of the developed averaged model can be evaluated comparing the thick solid line for the original model and the thick dashed line for the averaged model. While the starting permeability is slightly underestimated for a flux in stacking direction (lower angles towards $\varphi = 0$), the overall saturation behaviour is replicated accurately for all angles.

6 Conclusion

The proposed elliptic interpolation of the co-energy density produces convex functions for a stable and thermodynamically consistent simulation. The generated look-up table of the relative permeabilities $\mu_{r,i}$ can be easily imported in FE-software like COMSOL Multiphysics. While we showed the interpolation between perpendicular uniaxial measurements, the data of other directions could be also included in the construction of the contour surfaces. This might be particular useful for GO steel at low fields where the angle between the magnetic hard direction and the easy rolling direction is close to 60° as discussed in [9]. Our model can be applied to anisotropic materials with totally different permeability ranges as shown for GO steel and MSMA.

For laminated materials, with an increasing stacking factor the lamination becomes dominant compared to the anisotropy of the base ferromagnetic material. The overall behaviour of the laminate is dependent on the complete co-energy density w'_{lam} , in contrast to that, the permeability is dependent on the co-energy density w'_{an} in the ferromagnetic material. Even very high stacking factors f with only small gaps in the ferromagnetic material already cause a drastic change in the contour lines and surfaces of w'_{lam} compared to the material without lamination, while w'_{an} is less effected. The permeabilities calculated from w'_{an} are much smoother than those generated from the gradient of w'_{lam} and correctly replicate the high anisotropy.

The numerical validation shows a very high accordance of the fully modelled laminate and the deduced

homogenization rule. This opens many possibilities for efficient models of magnetic circuits without the need to mesh the thin gaps. For dynamic simulations, the described magnetic behaviour is not sufficient. Besides hysteresis, the electrical conductivity σ becomes important to estimate eddy currents. For laminated stacks, different models of σ are compared in [10].

References

- [1] S. Lopez, et al. Grain Oriented Steel Assembly Characterization for the Development of High Efficiency AC Rotating Electrical Machines, *IEEE Trans Magn.* 2009; 45(10): 4161–4164. doi: 10.1109/TMAG.2009.2023243
- [2] Y. Ushigami, et al. Recent development of low-loss grain-oriented silicon steel, *J Magn Magn Mater.* 2003; 254–255: 307–314. doi: 10.1016/S0304-8853(02)00933-2
- [3] T. Nakata, et al. Measurement of magnetic characteristics along arbitrary directions of grain-oriented silicon steel up to high flux densities, *IEEE Trans Magn.* 1993; 29(6): 3544–3546. doi: 10.1109/20.281224
- [4] D. Miyagi, et al. Development of Measuring Equipment of DC-Biased Magnetic Properties Using Open-Type Single-Sheet Tester, *IEEE Trans Magn.* 2006; 42(10): 2846–2848. doi: 10.1109/TMAG.2006.879144
- [5] A. Di Napoli, R. Paggi. A model of anisotropic grain-oriented steel, *IEEE Trans Magn.* 1983; 19(4): 1557–1561. doi: 10.1109/TMAG.1983.1062582
- [6] J.C. Sabonnadiere, G. Meunier. Modelling Nonlinear Magnetic Materials for Field Computation, *J ELECTROMAGNET WAVE.* 1990; 4(11): 1027–1054. doi: 10.1163/156939390X00357
- [7] J. Krause, Energy conserving Anisotropic Anhysteretic Magnetic Modelling for Finite Element Analysis. 2012; arXiv:1212.5163v1 [math-ph]
- [8] M. Schautzgy, U. Kosiedowski, T. Schiepp. 3D-FEM-Simulation of Magnetic Shape Memory Actuators, *Proc. of 2016 COMSOL Conf.*; 2016 Oct; Munich.
- [9] H. Wang, C.-S. Li, T. Zhu. Hard magnetization direction and its relation with magnetic permeability of highly grain-oriented electrical steel, *INT J MIN MET MATER.* 2014; 21(11): 1077–1082. doi: 10.1007/s12613-014-1012-8
- [10] J. Ziske, H. Neubert R. Disselinkötter. Modeling of Anisotropic Laminated Magnetic Cores using Homogenization Approaches, *Proc. of 2014 COMSOL Conf.*; 2014 Sep; Cambridge.

Simulation des Einflusses von Oberschwingungen auf die Spannungsqualität und Übertragungsverluste

Benjamin Jacobsen ^{1*}

¹ Professur für Energie- und Hochspannungstechnik, Technische Universität Chemnitz, Reichenhainer Straße 70, 09126 Chemnitz, Deutschland; *benjamin.jacobsen@etit.tu-chemnitz

Extended Abstract

1 Motivation

Die Versorgung mit elektrischer Energie ist eine Grundsäule für die Aufrechterhaltung der Wertschöpfung sowie der Innovationskraft von Industrienationen und ist damit von entscheidender Bedeutung [1]. Die Güte bzw. Unterbrechungsfreiheit der Elektroenergieversorgung wird mit dem Begriff der Versorgungszuverlässigkeit beschrieben. Dieser Begriff setzt sich aus verschiedenen Kennzahlen zusammen und ist nicht eindeutig definiert. Krämer und Westermann zeigen eine originelle und schlüssige Aufteilung der Versorgungszuverlässigkeit in Netzzuverlässigkeit und Spannungsqualität (als Produktqualität benannt) [2]. Bei der Recherche in anderen Quellen sowie bei den Bestandteilen der Netzzuverlässigkeit selbst nimmt die Spannungsqualität immer wieder eine entscheidende Rolle ein. Damit kann geschlossen werden, dass die Spannungsqualität eine elementare Grundlage zur Aufrechterhaltung der Versorgung mit elektrischer Energie ist [3]. Simulationen im Bereich der Spannungsqualität sind aufgrund fehlender Grundlagen und Methoden in der Regel nur vereinzelt zu finden. Dazu kommt das weit gefächerte Verständnis am Begriff der Spannungsqualität. Am besten lässt sich der Inhalt nach den Regeln entsprechend der DIN EN 50160 beschreiben, indem unter der Spannungsqualität die Parameter (u.a. Spannungshöhe und Oberschwingungsspannungsgehalt) der tatsächlichen Versorgungsspannung subsummiert werden. Innerhalb der genannten Norm werden wiederum für alle Parameter Grenz- und Sollwerte aber auch mögliche Ausnahmen definiert. Es handelt sich demnach bei der Spannungsqualität um ein Konglomerat an verschiedenen Parametern zur Beschreibung der sinusförmigen Wechselspannung. Einen weiteren Ansatz zur Beschreibung der Spannungs-

qualität leistet die englische Übersetzung. In der Regel wird der deutsche Begriff entsprechend seiner Bedeutung mit „Power Quality“ übersetzt. Diese Translation geht auf den Umstand ein, dass es bei der Versorgung mit Elektrizität weniger um die Spannung (im Sinne eines elektrischen Potentials), als vielmehr um die damit zu verrichtende Arbeit bzw. die aus diesem Potential gewonnene Leistung geht.

2 Methodik

Mit einer Grundgleichung der Elektrotechnik wird ein direkter Zusammenhang zwischen der normierten Spannung und der eigentlich benötigten Leistung hergestellt.

$$P = \frac{U^2}{R}$$

P ... elektrischer Leistung in Watt
U ... elektrisches Potential (Spannung) in Volt
R ... ohmscher Widerstand in Ohm

Der Widerstand wird in der Regel durch das Netz vorgegeben und ist damit nicht beeinflussbar, somit bleibt zunächst nur eine Beeinflussung der Spannung um die geforderte Leistung erbringen zu können. Es zeigt sich, dass die Bewertung der Netzspannung zur Gewährleistung der Bereitstellung elektrischer Leistung bzw. (wenn das Integral der Leistung über die Zeit gebildet wird) elektrischer Energie genügt. Bei der eigentlichen Bewertung der Spannungsqualität und einer daraus folgenden Simulation für verschiedene Netz- und Lastsituationen bilden sich nun hauptsächlich zwei Herausforderungen. Die erste besteht in der Schlussfolgerung der Güte der Spannungsqualität aus den einzelnen Parametern. Es existiert bisher noch keine Methodik, mit welcher die Spannungsqualität als einheitlicher Parameter (beispielsweise mit einem Wertebereich zwischen 0 und 1) bestimmt werden kann. Die Herausforderungen, die sich daraus ergeben wurden unter anderem von

Jasinski et al aufgezeigt [4]. Elphick et al haben gezeigt, wie schwer allein die messtechnische Erfassung und Bewertung der einzelnen Parameter der Spannungsqualität in Gänze ist [5]. Nourollah und Moallem haben sich (wie schon einige vor ihnen) darum bemüht eine Methode zur Zusammenfassung der Spannungsqualität in einer Kennzahl zu ermöglichen [6]. Alle diese Bemühungen haben gezeigt, dass die Spannungsqualität eine zu individuelle Kennzahl ist, als sie universell beschreiben zu können. Hierbei handelt es sich um ein Spezialproblem, welches in weiteren Veröffentlichung diskutiert und bearbeitet wird. Die zweite Herausforderung liegt in der Bewertung der einzelnen Parameter im Hinblick auf die Spannungsqualität. Um diesen Einfluss simulativ nachgehen zu können, sollte zunächst mit nur einem Parameter gearbeitet werden. Die Wahl hierfür fällt auf den Oberschwingungsgehalt, das hat vor allem zwei Gründe. Zum einen existiert für die Leistung- bzw. Energie zur Kompensation von Oberschwingungsspannungen (THD – Total Harmonic Distortion) bisweilen noch kein Markt (anders bei Regelleistung zur Frequenzhaltung oder Blindleistung zur Regulierung des Phasenwinkels), der bei der Simulation beachtet werden müsste. Weiterhin handelt es sich um ein eher lokales Phänomen, welches sich nicht netzweit (das bedeutet nicht nur innerhalb eines Netzabschnittes / einer Leitung) ausbreiten kann. Die Herausforderung bei der Simulation von Oberschwingungsspannungen besteht demnach hauptsächlich in der Simulation der Auswirkungen von Oberschwingungsströmen in lokalen Netzen. Quellen von Oberschwingungen (die vor allem durch Leistungselektronik dargestellt werden), senden keine Oberschwingungsspannungen aus. Durch ihre eher getaktete Leistungsaufnahme verursachen sie Oberschwingungsströme, die sich erst durch die Netzimpedanz zu Oberschwingungsspannungen ausbilden.

3 Ergebnisse

Ein Ansatz zur Bewertung des Einflusses von Oberschwingungen (und später auch von weiteren Parametern) auf eine mögliche zusammengefasste – universelle Spannungsqualitätskennzahl ist die Simulation der nötigen Energie zur Kompensation der Störbeeinflussung. Die Bewertung des Einflusses von Oberschwingungen auf die Spannungsqualität wird in einem ersten Schritt durch den Vergleich der Oberschwingungsverluste mit den Gesamtnetzverlusten realisiert. Die Simulation der Werte auf dieser Grundlage hat den Vorteil, dass zunächst besser beurteilt werden kann, in welchem Umfang eine Kompensation erforderlich bzw. sinnvoll ist, bevor nähere Kenntnis

zu einem (einheitlichen) Spannungsqualitätskennwert vorliegen.

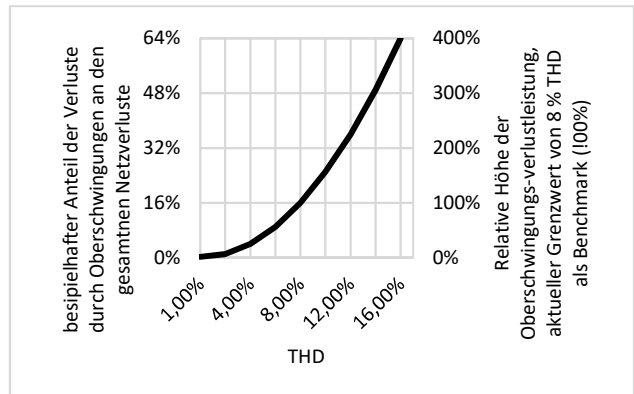


Abbildung 1: Abhängigkeit der Netzverluste vom Oberschwingungsgehalt (THD)

Abbildung 1 zeigt deutlich ein quadratisches Wachstum der Verluste durch Oberschwingungen bei zunehmender Oberschwingungsspannung. Es wird zum einen deutlich, dass die Art (bei Oberschwingungen als relative Größe in % der Nennspannung) sowie die Höhe (hier 8 %) der in der Norm vorgegebenen Grenzwerte nicht zu kritisieren sind. Des Weiteren kann deutlich erkannt werden, dass im unteren einstelligen Prozentbereich keine hohe Relevanz zur Kompensation von Oberschwingungen vorliegt. Anders ab einem zweistelligen THD-Gehalt, hier ist mit einer signifikanten Erhöhung der Netzverluste durch Oberschwingungen zu rechnen und es sollte zu einer Kompensation kommen.

4 Ausblick und Diskussion

Ziel weiterer Untersuchungen muss es sein, diese Erkenntnisse so zu verarbeiten, dass daraus Regelalgorithmen simuliert werden können. Dazu werden technische Möglichkeiten zur Kompensation der Oberschwingungen simuliert. Die Anlagen werden mit ihren Parametern darauffolgend in Beispielnetzen eingebracht. Anhand verschiedener Szenarien wird simulativ bestimmt, ob eine zentrale oder dezentrale Strategie bei der Kompensation verfolgt werden sollte. Eine dezentrale Kompensation könnte so weit reichen, dass an jedem Netzanschlusspunkt (beispielsweise jeder Hausanschluss) eine entsprechende Anlage zur Kompensation von Störbeeinflussungen installiert wird. Dezentrale Ansätze hingegen würden eine Kompensation beispielsweise an der nächsten Ortsnetzstation oder erst am nächsten Umspannwerk vorsehen. Zusammenfassend handelt es sich bei der Auslegung der richtigen Kompensationsstrategie um ein typisches Optimierungsproblem. Die Summe aus

Aufwand für die Kompensation und den Netzverlusten soll ein Minimum erreichen. Um dieses Optimum universell generieren zu können, soll ein Simulationstool geschaffen werden, welches in der Lage sein soll die optimale Oberschwingungskompensationsstrategie zu finden. Weiterführend soll diese Software für weitere Parameter der Spannungsqualität ausgebaut werden, was langfristig zu einer Minderung der Übertragungsverluste, sowie einer bedarfsgerechten Erhöhung der Spannungsqualität und einer höheren Versorgungszuverlässigkeit beiträgt.

Referenzen

- [1] Schwab A. J. Elektroenergiesysteme: Erzeugung, Übertragung und Verteilung elektrischer Energie, 5th ed. Berlin, Heidelberg: Springer Vieweg, 2017. [Online]. Available: <http://dx.doi.org/10.1007/978-3-662-55316-9>
- [2] Krämer M. and Westermann R. "Die Qualitätsregulierung im Strombereich aus der Perspektive großstädtischer Verteilnetzbetreiber," in Energiewirtschaftliche Tagesfragen, pp. 16–19.
- [3] Böske J. Zur Ökonomie der Versorgungssicherheit in der Energiewirtschaft. Zugl.: Münster (Westf.), Univ., Diss., 2007. Berlin: Lit, 2007. [Online]. Available: www.gbv.de/dms/zbw/547565976.pdf
- [4] Jasinski M. Sikorski T., Kostvla P., and Borkowski K., "Global power quality indices for assessment of multipoint Power quality measurements," in ECAI 2018 - International Conference – 10th Edition.
- [5] Elphick S. Ciufo P., Drury G., Smith V., Perera S., and Gosbell V., "Large Scale Proactive Power-Quality Monitoring: An Example From Australia," in IEEE TRANSACTIONS ON POWER DELIVERY, VOL. 32, NO. 2, APRIL 2017, pp. 881–889.
- [6] Nourollah S. and Moallem M. "A data mining method for obtaining global power quality index," in pp. 1–7.

Die vorgestellte Forschung wurde durch das Projekt GRIDS – Grüne Energie in industriellen Verbünden unterstützt



Modeling of mechatronics system on example on window regulator: conceptions and troubles

Sergey Petkun¹

¹System Integration Door Modules, Brose Fahrzeugteile SE & Co.KG,Bamberg, Berliner Ring 1, D-96052 Bamberg; sergey.petkun.brose.com

Abstract. The mechatronics systems are inalienable part of present industrial progress. They are extended from small micro devices in nano range up to huge systems like ocean liners. Common for such systems is presence of four important components: mechanical system providing energy from electrical motors managed by an electronic with software control. Mechatronics is technology combining electronics and mechanical engineering. The microsystems (MEMS) are excluded from our consideration. Mechatronics systems will be restricted to dynamical systems in this paper. The movement is necessary part of dynamics. The company, manufacturing the all four components of mechatronics device, is absolute exception in industry. The most of companies are suppliers of one or two subcomponents to the whole mechatronics system. Every supplier takes a focus on own product and its domain. The simulation is not required at all if device works fine after assembling. This ideal case happens not very often because of growing complexity. In reality, there are a number of loops to set device in robust working state. At present, the modern progress is very paradoxal: the systems are becoming more complex and at the same time, they are designed in shorter period. The system simulation can help to cope with these difficulties. The problems, connected with the system simulation, are considered in this paper from industrial point of view. Electrical window regulator will be analysed as an example of mechatronics system. It has all necessary components like mechanical part with motor-controlled electronics with software. The simulation system BroSAnT will be given as an example of realization the system analyse.

Introduction

Today, the progress of industry is characterized with high dynamics. Every domain has its own inner development. Electronics is on the way of quantity expansion (more CPUs, memories and so on) and minimization of

sizes (microminiaturization the chips). The mechanics tries to optimize the weight of construction (new forms and materials). The software offers more functionalities. The drives and motors are offered in very wide choice in power, sizes and prices. All these processes run under the pressure of costs and time. All these effects influence the process of development the mechatronics systems. Plenty questions arise during the development of the system modelling. Every simulation is to answer at certain special questions. The many questions can be formulated to whole system during the development. At the same time, there are relatively small number among them, which are technically correct formulated. The main attention to system is in its functionality. The functionalities of the product are strategical moment in business. The management makes the decisions about the desired functionality. The skill of management is concentrated in economic and financial area as a rule. That is why, they formulate the correct technical questions very seldom. The process of globalization leads to growth the managing structures and their hierarchy. The technical knowledge and experience remain at the bottom of this hierarchy. The opinion of experts has the small weighting on taking the decisions. Technical understanding is the first what is needed for mechatronics simulation. Wherefore, the target group of mechatronics simulation is engineering staff for integration system from supplied subsystems (components). Usually this staff is called application team. The assembling of system is the right place for system simulation of mechatronics systems. But there is no time for simulation as a rule in project schedule. The development of components is already almost finished at the phase of system integration. Every change of the component prototype leads to increasing the costs of product. The last member in the development chain of the system, namely the integration (application) engineer, is the judge for what must be changed for optimal system operation. But this engineer is not decision-

maker. The economic competition and market pressure give rise to shifting the decision-making to purchasing department. The decisions of dominated financial sectors often lead to some technical problems which arise later during the integration phase or yet even later during recall campaigns.

The industrial planning process goes out from the point of view that all components are designed properly for its functional purposes. The stumble stone is here the possible problem which can occur during the integration subsystems. All components meet all purchaser's requirements and despite that, the device does not operate properly. The system simulation would be helpful on phase of the formulating the proper requirements for components. However, it is not clear enough who should provide such simulation for the integration engineering at this phase. The business of suppliers is already finished after delivery their product. As a rule, the supplier is not required to deliver a model of its component, which could be used for simulation purposes. The modelling on system level can be used for different purposes. An excellent introduction in the modelling of mechatronics devices can be found in [1].

1 Mechanical part

The stress, strain, stability, durability and endurance are in focus of interests the mechanics during the development of mechanical devices. The creation of mechanical construction begins from CAD (**C**omputer-**A**ided **D**esign) drawing. It is very easy to produce an animation of a moving object based on its geometry at this phase. For mechanical engineering the animation of motion is the first step of understanding functionality of the system. The important questions arise already in initial phase of design: tolerances, choice of materials, the geometry variations under wear-out, temperature Influences and so on. The engineer gets answers for these questions partly from test, partly from own experience and simple calculations, sometimes from literature and, of course, from the simulation department. There are a number of complex tools, which are trying to answer on questions of interest.

Very often, CAD drawings are the initial input as source for mechanics simulations. Engineer uses the geometrical drawings for creation of simulation model. The next step is to fill CAD geometry with physical content of material properties. After that, the different kinds of element connections (joints) have to be cre-

ated. At this manner, the model for MBS (**M**ulty**B**ody **S**imulation) is created. ADAMS(MSC Software) and SIMPACK(Dassult Systèmes) are typical examples of MBS and are used at early development phase. There are many simulators in MBS group from commercial up to free like MBDYN. All of them produce an excellent animation of system motion. Often, they combine the analytical approach with numerical FEM (**F**inite **E**lement **M**ethod). The model parametrization is most important and difficult part during its building for modelling of dynamics. Parameter identification is the weakest place of simulation. For example, translational and rotational stiffnesses are needed to set in all 3 dimensions for free joint. It is needed already to determine 6 unknown parameter for only single. Next step is to bring in modelling object the losses (for example, friction) in form of acting external or internal forces. There is not sure information sources for it in many cases. All this information is not really needed for kinematic animation, which is frequently mixed up with the result of dynamical motion. The complexity of inner life of the tools from one side and requirement for easy usability of it from another one, leads to creation a user-friendly GUI (**G**raphical **U**ser **I**nterface). All computational mechanism is hidden behind this user interface. The inexperienced user is very often overloaded by the amount of information necessary for simulation. The geometrical preparation (proper mesh) of the model is no single task of simulation engineer. The user of such tool has to provide a lot of information about external environment: external forces for stress and strain simulation (boundary conditions). Wherein, it is oft forgotten, that the quality of output information depends not only from tool but also from quality of input information. The model will be simulated after successful geometrical preparation, filling physical content and forces in it. The result analysis is required no less experience than task of preparation for the input data. Several models are arisen at the end of development the mechanics. They bring the answers on various problem formulations. As a rule, all of them are pure analog.

2 Electrical part

The electronics task in system simulation seems only easier as mechanical one. The correct management of motor requires proper functionality under unlimited combinations of conditions and effects. The most important of them are thermal and climate influences. The

manufacturing tolerances are strongly connected with prices. It has also to be taken into account. The temperature appears two times as an object of interest. The thermal behaviour of electronics components is studied in different environment conditions and operation modes. The diverse simulations are created during the development of the controlling device. The focus of interests is shifted in direction of robust operation in the prescribed environment. The new field of interest, in comparison with mechanics, is EMC (ElectroMagnetic Compatibility). 3D-models are used for estimation of electromagnetic radiation and temperature distribution. The functionality of electronics component is studied at the beginning of design process by means of simulation of the electrical schematic. The thermal and EMC models of electronics are analog almost always, but functional one consists from both analog and digital parts. The animation is fully absent as part of result analysis in simulation world of electronics. There are many models of electronics components, which are based on its behaviour. The physical complexity of electronics cannot be described by means of simple analytical mathematical language. The progress of electronics has gone more empiric than mechanics. The creation of the stable working electronics devices is not possible without clear understanding of the functionality of its elements. The supplier provides already the most important parameter of electronics part for the functional modelling. In contrast of mechanics, the absence of animation during the development of electronics played positive role for electronics progress. The same as in mechanics development, many models of electronics are created for different purposes and types. There are analog, digital and mixed models.

3 Software part

The smart thinking is a very important part of modern mechatronics system. It is packaged in form of software (so called firmware) in electronics chip. That is why that the software in mechatronics sense is also part of modern electronics. The irony of progress consists in that the "brain" of mechatronics system is fully digital in contrast to real world which is analog (at least macro world). The peculiarity of software part is deceptiveness of easy, cheap and fast development. But it is a part as complex as the two others. Only its distribution is easy and cheap. As a rule, the software is relatively unproblematic to adapt for usage in simulation. The

proficiency is in that the "small adaptation" of software should behave the same like original one. The software plays a very important role in mechatronics simulation. It is "an incorrupt judge" to the suitability and quality of the rest of the simulation. The software knows nothing about the rest of the simulation. It can be embedded in real hardware or only in reproduction of its reality. If the software shows the same behaviour in simulation as in real hardware, it means that the modelled rest of system is proper for this deepness level. The great advantage of the software part in comparison with rest parts, that it can be embedded in whole simulation in form of shared library like *dll* or *so*. The problem of know-how protection is usually solved with it.

4 Drive part

The energy source is needed for dynamics and namely for motion. The electrical motor is used as the source of energy for most of mechatronics systems. There are huge range of motors from micro up to giant size. The great range of different constructions is available nowadays. The transformation of electrical energy in mechanical one is common intended purpose of all motors. Series of simulations arise during the development of motor. The motor models are developed and used during development of mechanics or electronics also. Peculiarity of the development of motors is the simulation of magnetic field dynamics for proper choice of its topology.

The development of this mechatronics component is closely connected with both domains: in mechanics (dynamical movement, friction) and in electronics (current-voltage characteristics, thermal behaviour). Constructive combination of the motor with some gears leads to conception of the drive. The end user is needed power from supplying energy. The power can be got from motor through rotational velocity. For example, the drive unit of electrical window regulator consists from electrical motor combined with worm gear. This constructive combination brings the properties named self-locking. At the end of drive development, a few different models are available. Every of them answers on specific class of questions. There are both analog and digital models, 1D and 3D.

5 Simulation integration

From above mentioned, the idea comes about the integration already available simulations into some over-all system simulation. This concept is named FMI (**F**unctional **M**ock-Up **I**nterface). It is some kind of generalization of the concept of co-simulation, where two or more simulators run in parallel simultaneous. They exchange information with each other during running process. The main idea of this concept is to organize the interface for information exchange between dynamical models of different components. Another concept consists in attempting to describe as many as possible subsystems in frame of the same language. It gives possibility to use a single simulator for evaluation. There are various trends going out from different domains. The electronics gone way to extend the available language on integration of mechanics in it. The VHDL (**V**ery **H**igh **S**peed **I**ntegrated **C**ircuit **H**ardware **D**escription **L**anguage)) was extended to VHDL AMS (**A**nalog **M**ixed **S**ignal) on this way [2]. The language SIMSCAPE(Mathwork) was come from software and algorithm sector. It was wish to stay in environment SIMULINK(Mathwork). The language MODELICA was initiated from mechanics and drive domains for possibility to analyse the whole system [3]. There are few of simulators both commercial and free based on MODELICA language. All these languages are based on principles network simulation (nodal analysis). The main purpose of this languages is to find some small set of basic (fundamental) models with aim to enclose as many as possible mechanical effects. The main device classes should be built by means of these properties. It is like searching the letters for alphabet to allow writing as many as possible words by means this language. The set should not be very big but versatile and powerful. All models can be divided in two groups "fundamental properties" and "constructive solutions". Example of last one is rack and pinion model, which serves for transformation from rotational to translational movement and vice versa. Classical example of fundamental model is model of mass, which is described the effect of inertia. Property of elasticity is modelled with ideal spring and so on. The simulators are based on network concept (node and mesh principles) The base laws for electrical domain are Kirchhoff's laws. Kirchhoff's law for nodes: **The algebraic sum of ingoing and outgo-**

ing currents (I_i) in any node is equal to zero.

$$\sum_{i=1}^n I_i = 0$$

Kirchoff's law for mesh: **The sum of voltage drops (ΔV_i) around any closed loop is zero.**

$$\sum_{i=1}^n \Delta V_i = 0$$

Analogue of Kirchoff' node law is in mechanics D'Alembert's principle of inertial: **The sum of all forces inclusive inertial forces (F_i) in node is equal zero.**

$$\sum_{i=1}^n F_i = 0$$

The mesh Kirchoff's law has analogy in mechanics with principle of continuity the mesh: **The algebraic sum of all deformations (Δu_i) around any closed loop is zero** (the mechanical chain is not permitted to be disrupted).

$$\sum_{i=1}^n \Delta u_i = 0$$

The language VHDL AMS is standardized. But there is not defined in language standard the force flow direction. The model of mass can be defined as

$$F^i = -m * \ddot{x} \quad \text{or} \quad F^i = +m * \ddot{x}$$

It does not matter which definition is taken. They are valid both and can have a place. There is very unusual treatment of inertia properties in [4]. This model describes only action of inertial force F^i the mass m in the node by an acceleration \ddot{x} . The consequences of this choice have at once influence on choice the sign for the elasticity force in spring and rest of mechanical elements dealing with force. For example, ideal elasticity properties should have the cooresponding to mass model sign

$$F^e = -k * \Delta x \quad \text{or} \quad F^e = +k * \Delta x$$

Where F^e - elastic force, k - spring stiffness, Δx - its deformation relative to unstressed state. The choice of sign (force flow direction) is in language standard VHDL not prescribed. That is why it is not reasonable to use VHDL AMS mechanical model without knowing which direction is chosen. It is relatively easy to determine chosen direction for two terminal model. But it is not so trivial it to determine for model with many terminals. At present, there are a wide range of mechanical models with one or another direction. The both variants

are used actual. Models from electrical sector are not affected by the sign choice on mechanical forces. In MODELICA the flow of force in flanges is clearly defined. It is useful to clear out requirements for simulation from the target user group sight. It brings more clarity for way the creation of the system simulation for whole mechatronics device.

6 Requirements for system simulation

The target group of potential users of system simulation was already defined as integration engineering team. To make the whole system to more robust and safety, they need the virtual prototype of their system for successful and effective working. The wish of integration (application) team is to build the whole system from "black" boxes: mechanical part, control electronics with software, electrics (drive). The simulation building should be reflected in this approach: the parameter which are known from real system combination should be reflected in the simulation parameters. Only developer of subsystem has deepest knowledge about it's at most an can estimate on expert level quality of behaviour own subsystem especially in case of fail problem. But the behaviour is their component in whole system is not so well known for them, because they develop component stand-alone without rest environment. They are indifferent to whole system. Only proper behaviour their part is interested and important for them.

System integration is the last member in development chain. The time pressure is very probable at the end of project schedule. It leads to demand on restriction the simulation time. Many days-simulations can be acceptable on component development phase. But it is unacceptable during integration period. The parameter set should be famous to integration engineers from their daily work. The integrators don't need very precise information from component's sector. Some specific requirement comes from our experience for long period: very important to organize the clear and effective simulation analysis of their results. The many different situations can be simulated for the virtual device with relative quick calculation time during short period. It can come to information chaos if the system of result analysis is not well organized.

There are the big differences on usage simulation tool in different component's domain. It is connected with complexity of tools, duration of evaluation time,

post process proceeding and so on. In mechanics domain the simulation activity is separated from construction sector. The dominant method of calculation is FEM in mechanical domain. Usage of this method needs the big and complex process of preparation the input data, long calculation time and laborious post process for analysis of the results. It leads to natural separation simulation engineering from construction one. Table calculation tool Excel is used frequently for their own calculation. The situation is differed in electronics sector. The FEM approach is used as in mechanics for some complex simulation how heat distribution in space (3D) of electronics plate. But functional simulation is made by electronics developer itself. The simulation usage is separated from developer only partly in software sector. Sometimes the developer himself simulates its part with SIMULINK or others simulator. Different simulations are in use by electrics (drive) sector for choice topology of motors, prognoses of acoustic and thermal behaviour and so on. Almost all of them are based on FEM calculation. Their simulation usage is separated from direct developer. They have own simulation teams.

It happens very seldom that the first integration of all subsystems is acceptable at once for series production. The integration loops are need for optimization of interaction the system parts with each other. Simulation of whole system can be very useful to determine what must be changed and why. The subsystem suppliers are usually locally distributed. That is why, the network accessibility to simulation is strong required.

7 System dynamics

The dynamical interaction between subsystems during the movement is absent on subsystem level. Especially it is important in situation when the system state is changing. It can be changed from movement to stop or start up process. The most of dynamics is concentrated in start-stop phases. The next important point is connected with digital behaviour of simulation system. The pure "zero" does not exist in simulation the same as in reality. Classical analytical oscillation model "spring-mass" with damping is represented by equation

$$m \cdot \ddot{x} + d \cdot \dot{x} + k \cdot x = 0 \quad (1)$$

where mass m oscillates on the spring with stiffness k and damping d . This model reflects many cases from

reality very good. The only problem is with it that the mass in this model is never stopped according to this model despite presenting of damper in system. Usage of the classical model for dry friction (2) leads frequently to numerical oscillation [5]. The concise mathematical definition of dry friction is not simple to realize in frame of simulation language. The numerical stop state means zero velocity. The stop (zero-velocity state) depends on computers, tools, used algorithms during computation and many other reasons. By this way the new parameter is appeared in models something like "if velocity is less as 10E-15, it means that the velocity is equal zero". It is the usual way of behaviour modelling.

$$F^{fric} = \begin{cases} -k \cdot N \cdot \frac{\dot{x}}{|\dot{x}|}, & \text{if } \dot{x} \neq 0 \\ -F, & \text{if } \dot{x} = 0, |F| \leq k \cdot N \\ -k \cdot N \cdot \frac{F}{|F|}, & \text{if } \dot{x} = 0, |F| > k \cdot N \end{cases} \quad (2)$$

where F is the resultant of all forces, other than frictional ones, N - magnitude of normal (pressure) force, k - friction coefficient and \dot{x} - velocity of point the applied friction force.

Dynamics will have more and more place in all mechatronics system in future. It is consequence of economic competition in the world. The mechanical part is optimized for minimization of energy consumption during movement. It lead to reduction of losses in motion, what means the reducing friction forces. The second term in (1) damping d is nothing else the friction, which is dependent from velocity of motion. The less damping coefficient d is this differential equation, the more dynamics is in system mass-spring.

The friction in mechanics (losses) is negative feedback from point of view of the system theory. It is well known, if the smaller negative feedback is in system, then the less stable system is. All what is won from smooth-running system (smaller energy source, weight), will be lost in fight for system stabilization. Not the only mechanics struggles with friction. In electronics the analogue of friction is heat. The thermal management plays important role in usage of electronics. The classical mechanics has great progress for ideal case - without losses like friction. All analytical methods in mechanical science is functioned fine so long the friction forces or thermal effects are absent. But main friction laws (Amontons', Coulomb's) are empirical, however. Drive of window regulator has motor armature which is the worm of worm gear at the same time. Worm gear is used in car window regulator for po-

sition keeping of glass in case of switched off electrical motor. However, the price for it, is immolation the efficiency of drive. The self-locking of worm drive needs not the less than 50% of losses. All these losses are friction. The simulation in mechatronics without considering friction at system level has no practical sense at all. The implementation of mathematical description for friction (2) in simulation language represents the challenge for every developer [6]. The model should reflect not only mathematical sense but also be a numerically robust. There are a lot of other thing to do by it beside integration subsystem in whole system. They are activities like validation of simulation quality, constant update of libraries of subsystem, educating the subsystem supplier on effect of the embedding their component in whole system. The requirements for models are permanent changed with growing understanding of the system. It is useful to keep in mind, that any model is only a reflection of some reality side. By strong declination from initial problem, it can be occurred that the model is out "it's working range". New model or extension for old one is asked for new problem formulation. It is important always do not forget the "golden rule": the model should be lightweight as possible, but it has to consist all necessary properties for answering on questions to simulation. The verification and validation of subsystem models belong also to activities of system region, because of the model developer has not the complete test environment. It is very important not lose the sense of system view despite constant wishes for its extension. There are domain specific tools exactly for these purposes. The building of the system simulation requires the right system vision, which based on experience of developer and close cooperation with component designer.

8 BroSAnT like example of system simulation

BroSAnT is acronym from **Bro**se System Analyse Tool [7]. The electrical **Window Regulator (WR)** for automobile is one among the products of company Brose. This regulator is perfect representative of mechatronics system. The physical drive is motor combined constructive with worm gear. The BroSAnT is ordinary client-server service for whole company worldwide. Of course, the service is accessible for the users having proper access rights.

Almost any mechatronics system is potential dan-

gerous for user. The uncontrolled energy flow can damage the human. The WR is not exclusion from this rule. To provide safety usage of electrical WR, there is requirement for protect the human against pinch by means of window glass in domestic legislations. These legal requirements differ a little bit from country to country. But in principal, it is the same by all: electrical WR with automatic lift up function must have anti-trap protection against damaging of human under all conditions. A contact jam is allowed. But maximal jam force is restricted to 100 Newtons. The term "all condition" is more as 3D test space: (climate ($-40^{\circ}\text{C} \Rightarrow +80^{\circ}\text{C}$), working ($9\text{V} \Rightarrow 16\text{V}$) - testing ($10\frac{\text{N}}{\text{m}}; 20\frac{\text{N}}{\text{m}}; 65\frac{\text{N}}{\text{m}}$), time ($\text{now} \Rightarrow \infty$)). In addition to, the measure gauge can be place at any placed on window glass. That is why, it is very important to have virtual tool for preliminary testing of setting of firmware. The focus of simulation was an maximal force reached during clamping before reversing of the window glass.

The building of virtual WR reflects the real development process: choose the mechanical variant of WR, drive, electronics and firmware. Some input blocks are also added, to tune the measure and working condition. There is possibility to set measure gauge position on glass, stiffness of it. The glass geometry was considered through setting the action direction the gauge. The whole switch sequence and battery voltage are in input block of working condition, because the firmware was embedded with minimal adaptation. Additional information about the door of car and mass of glass was placed in separate input block. The choice of input information was minimized only for indeed necessary. The task is sending on to server after creation the WR with chosen parameters. Server sends back the simulation result of transient simulation to client after calculation of given task,. All models were developed in VHDL AMS. The simulator is SABER(Synopsis) at present. But thanks, VHDL AMS standardization can be changed on some another simulator. It is clear that mechanical engineer has not so deep knowledge about drive, electronics and or software. That is why concept of model libraries of components was realized. For example, the drive choice is based on name of drive manufacture and it's model. The simulation time is about the real time. It is very satisfactory for end user.

If other questions are set to simulation, it can lead to some other tool and modelling language. The process of anti-trap function remains, but the problem is state a little bit another like that "Which influence does the

glass tiling have on clamping process?" Another tool was chosen for answering on this question. More detailed glass geometry information was needed for this problem. The simulator was chosen MATHEMATICA (Wolfram Research Inc.) with its own language: WL (Wolfram Language):. The information about this modelling is in [8]. In the first simulation [7], the most important was distribution the kinetic energy between participants of motion. The friction and mass distributions become the leading role in second simulation. The simulation process remains the same despite usage of different simulators (SABER and MATHEMATICA). It is based on the server client concept. All information, which the simulator is needed, is keeping on server. The client knows model only through its parametrization. For end-user the model is only named parameter set. Such simulation has great potential, because yet on development phase, the different domain can test some future features which are too expensive to realize in reality or at all not possible. If all subsystems are inside single company, how it was in BroSAnT case, it is easier to get information and feedback from the subsystem supplier. But it is more exclusion as rule.

The parametrization of model is very significant for simulation building according this concept from one side. From another one, essential knowledge about the subsystem is very important. The parameter set says very much about what is the most important for component from view as participant in system cooperation. Some companies are active by simulation usage, other make accent on testing methods or rely on own experience. The companies, which supplied such parts like seal, simulate by means of FEM the elastic properties to get the necessary pressure profile for sure sealing. But there is no simulation activities by them during the time that glass goes up or down. The friction, which is the most responsible for dynamic behaviour, is up to now not in focus of their interest. The frictional properties of seal are very strong dependent from climate, thermal and assembling conditions. It makes the formulation of problem more difficult.

If only single parameter is in input unknown, it can be found on the base of comparison simulation result with measure some value (parameter identification process). Unfortunately, the case single unknown parameter is more exclusion as rule. It is needed high proficiency to proper model parametrization.

Measuring methods, in anti trap actions, play very important role and have the big influence on process

of pinch itself. Real measurements can be made if only the whole object exists already as real hardware. The changes in technological process of manufacturing or assembling the door can lead to new measuring cycle, because of they serve as base for firmware calibration.

There are almost always the both analog and digital parts in the system simulation of mechatronics device. VHDL AMS simulators have two cores: one for digital and another for analog evaluation. The language WL gives the possibility to proceed the events in it. But the combination of analog and event parts together is task of model developer.

9 Summary

The simulation of mechatronics devices requires the using system of approach to problem. An universal simulation does not exist, because of every simulation delivers the answers on technically correctly formulated questions. The subsystems from different domains are necessary "bricks" to build system simulation. These component models are also dependent upon formulated problems. The features, which are interesting for one problem, can be not useful at all for another

There is big gap between modern engineering and industrial science at present. The complexity of tools makes the user full helpless by usage of it. It would be developed the universal simulation tool from point view of the tool developer. In the same time, the main task the simulation engineer is quickly to give answer at industrial questions but no rather investigate of power of the tool which can do all. The trust to "universal" tools is inverse proportional to complexity of tool. The great example for it is progress on development of operational systems for PC. The mechatronics device is the best possibility to learn how can be coped the complexity of real system.

The most important rule of system simulation is "all what for problem is needed and nothing else!". Consequences of this rule is clear separation the interests on level subsystem and system. Only integration engineer can formulate correctly the requirement to simulation on system level. The questions like who and where should such simulation be created, maintained are remaining open in modern industrial structure. It can be seen the trend and interest to system understanding, but every domain has its accent at own product. The constantly rising complexity of integrated parts in system leads to growing interest to technical analysis on system

level. The system simulation can help really by that.

The system simulation is useful at early development phase to assure the concept. The system simulation will be having place in near future at some service companies offering the testing of mechatronics concepts. The certification of components of mechatronics system will be reason for supplying the models of subcomponent of system level. It will be not reasonable to start the mechatronics project without virtual testing for proper functionality of device. Such approach is already approved in domain of electronics. To sell the subcomponent, a company will be forced to deliver functional model at system level. The such testing is not full warranty of project success however it is better protection of technical solution than it is happened at present.

More dynamics brings essential changings in methods of analysis the system. "Analysis of curves" will replace the "analysis of numbers". For effective system analysis, it is required more analogies between electronics and mechanics domains. It will bring the understanding of system at more higher level.

References

- [1] G.Pelz *Mechatronic Systems: Modelling and Simulation with HDLs* ©John Wiley & Sons, Ltd, ©2003 , DOI:10.1002/0470867906
- [2] P.J. Ashenden, G.D. Peterson, D.A. Teegarden *The System Designer's Guide to VHDL-AMS* Morgan Kaufmann Publishers Inc. San Francisco, CA, USA ©2002
- [3] P. Fritzson *Principles of Object-Oriented Modeling and Simulation with Modelica 3.3: A Cyber-Physical Approach*, Wiley-IEEE Press. 2015
- [4] Malcolm C. Smith *Synthesis of mechanical networks: the inerter*, Automatic Control, IEEE Transactions on, Vol. 47, No. 10, pp.1648-1662, Oct., 2002
- [5] A. Fidlin *Nonlinear Oscillations in Mechanical Engineering* ©Springer-Verlag Berlin Heidelberg 2006, ISBN-10 3-540-28115-0
- [6] Armstrong-Hélouvry B., Dupont P., Canudas de Wit C. *A Survey of Models, Analysis Tool and Compensation Methods for Control of Machines with Friction*, Automatica, Vol30, No.7, pp1083-1138, 1994
- [7] S.Petkun, Xin Zhao *BroSAnt – an Example of the Mechatronical Way of Thinking*, ASIM Winthertur June 2011, SNE22(3-4)-12/2012, pp181-186
- [8] S.Petkun *Dynamic behaviour of power window regulator system with VHDL AMS*, 24th Symposium Simulationstechnik, ASIM 2018, pp7-16, Hamburg

Rechenzeitoptimierte Modellierung nicht-linearer physikalischer Prozesse mit Surrogate Modellen zur Anwendung in echtzeitfähigen Optimierungsverfahren

Marian Göllner^{1*}, Or Aviv Yarom¹, Jannis Fritz¹, Xiaobo Liu-Henke¹

¹Institut für Mechatronik, Ostfalia Hochschule für angewandte Wissenschaften,
Salzdahlumer Str. 46/48, 38302 Wolfenbüttel; *mar.goellner@ostfalia.de

Abstract. Auf Grundlage eines nicht-linearen Modells der Regelstrecke eines Intralogistik- Transportsystems „S-Mobile“ wird in diesem Beitrag ein optimierender, nicht-linearen Polvorgaberegler auf Basis der Jacobi-Matrizen entwickelt. Bedingt durch die nicht-linearitäten, vor allem der trigonometrischen Funktionen innerhalb des nicht-linearen Streckenmodells und damit auch in den Jacobi-Matrizen, wird der Einfluss auf die Echtzeitfähigkeit untersucht. Mit der Landau Notation der Algorithmen-Komplexität der zugrundeliegenden Berechnungsalgorithmen der Grundfunktionen, wird eine Methode zur Prädiktion der zu erwartenden Berechnungszeit vorgestellt. Ein Algorithmus, welcher ein heuristisches Modell als Surrogate-Modell der nicht-Linearitäten so trainiert, dass sowohl die Abweichung zu der Referenz klein als auch die Algorithmen-Komplexität insgesamt gering ist, wird entwickelt.

Einleitung und Motivation

Zur Verbesserung der dynamischen Eigenschaften von Gütertransportsystemen und deren Manövrierfähigkeit wurde an der Ostfalia Hochschule für angewandte Wissenschaften ein neuartiges Intralogistik Transportsystem „S-Mobile“, als ein hochdynamisches und unteraktuiertes System mit sphärischem Elektroantrieb, mechatronisch entwickelt. Da dieses unteraktuierte Mehrkörpersysteme nicht-minimalphasig ist und daher bei Arbeitspunktwechseln nicht-linearitäten aufweist und sogar instabil werden kann, ist die Regelung eine technische Herausforderung die nicht mehr mit konventioneller Regelungstechnik gelöst werden kann. Das in [1] sukzessiv entwickelte und in [2] vorgestellte nicht-lineare, hierarchische Regelungssystem basiert auf der

Berechnung der Systemzustände mittels kontinuierlicher Nachführung eines linearen Ersatzmodells auf Grundlage der Jacobi-Ableitungsmatrizen der nicht-linearen Systemdynamik. Die Approximationsgüte ist für die Funktion des Regelsystems maßgebend, erfordert allerdings eine dementsprechend exakte Nachbildung die zu Konflikten mit der Echtzeitfähigkeit des Reglers führen. Damit das Regelsystem unter Echtzeitbedingen eine optimale Zustandsregelung durchführen kann, soll in dieser Arbeit eine allgemeine Methode zum finden eines Surrogate Modell hergeleitet werden.

1 Modellbildungsprozess

Im Rahmen dieser Arbeit wird unter anderem die Modellbildung des nicht-linearen Streckenverhaltens besprochen. Nachfolgend sollen alle grundlegenden Schritte für die spätere Implementierung des nicht-linearen Verhaltens in den Regler dargestellt werden.

1.1 Methode

Der dazu notwendige Modellbildungsprozess (vgl. Abbildung 1) basiert auf dem realen System, welches gemäß der Aufgabenstellung reduziert bzw. vereinfacht wird, sodass sich ein physikalisches Modell ergibt. Dieses wird mithilfe physikalischer Gesetzmäßigkeiten in ein mathematisches Modell überführt, welches wiederum bspw. in Form von Signalflussplänen im Rechner abgebildet und mithilfe von CAE-Werkzeugen und entsprechender Numerik simuliert werden kann.

Der Modellbildungsprozess umfasst zudem Messungen am realen System, um zum einen die Parameter des mathematischen Modells zu identifizieren und

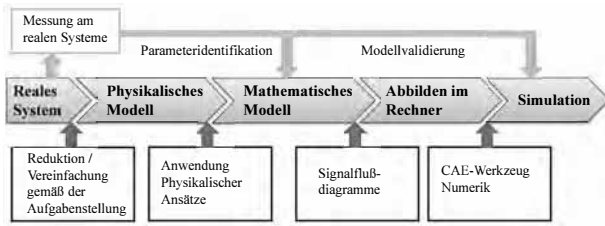


Abbildung 1: Modellbildungsprozess

zum anderen die Simulation zu validieren. Weitere Informationen sind für den planaren Fall in [3] und für die holistische driedimensionale Betrachtung in [1] gelistet.

1.2 nicht-lineares Zustandsraummodell

Ein Bereits in [2] für das S-Mobile auf Basis des Euler-Lagrange Ansatzes hergeleitetes Dynamikmodell wurde in die allgemeine Matrizenschreibweise überführt und dient als Grundlage der nachfolgenden Betrachtungen. Die nachfolgende Gleichung zeigt das verallgemeinerte Dynamikmodell mit Reibmodellierung aus [4] zur Berücksichtigung der Kraftübertragung.

$$\underline{\underline{M}}(\underline{q}) \cdot \ddot{\underline{q}} + \underline{\underline{C}}(\underline{q}, \dot{\underline{q}}) \cdot \dot{\underline{q}} + \underline{\underline{G}}(\underline{q}) = \underline{\underline{Q}}(\underline{q}) \cdot \underline{u} \quad (1)$$

Die folgenden Abbildung 2 zeigt die allgemeine Struktur des Dynamikmodells der Strecke.

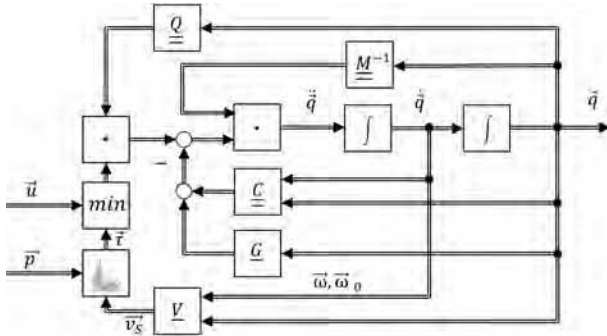


Abbildung 2: Modell der Strecke

Dieses Modell besitzt die Eingänge:

- $\underline{u} \in \mathbb{R}^n$ Krafteingangsvektor
- $\underline{p} \in \mathbb{R}^n$ Vektor der Pressungen

die Zustandsvektoren:

- $\underline{q} \in \mathbb{R}^m$ Zustandsvektor

- $\underline{v}_s \in \mathbb{R}^n$ Schlupfvektor
- $\underline{\tau} \in \mathbb{R}^n$ Kraftübertragungsvektor

und die charakterisierenden Matrizen und Vektoren:

- $\underline{\underline{M}} \in \mathbb{R}^{m \times m}$ Massenmatrix
- $\underline{\underline{C}} \in \mathbb{R}^{m \times m}$ Coriolis- oder Zentrifugalkraftmatrix
- $\underline{\underline{G}} \in \mathbb{R}^{m \times m}$ Gravitationsmatrix
- $\underline{\underline{V}} \in \mathbb{R}^{m \times m}$ Matrize der Geschwindigkeitsvektoren
- $\underline{\underline{Q}} \in \mathbb{R}^{m \times n}$ Eingangsmatrix der eingprägten Kräfte

Der Eingangsvektor \underline{u} wird über eine Minimalbedingung jedes einzelnen Vektorelements i definiert. Aus [4] wird ersichtlich, dass entweder direkt das Antriebsmoment der Aktuatoren \underline{u} oder aber eine maximal übertragbare Kraft $\underline{\tau}$ in das Modell weitergeleitet wird.

$$\forall (u_i, \tau_i) \in (\underline{u} | \underline{\tau}) :$$

$$\underline{u} = \min_{i=1 \dots n} [u_i, \tau_i] = \begin{cases} u(i) & u_i \leq \tau_i \\ \tau(i) & u_i > \tau_i \end{cases} \quad (2)$$

Der Kraftübertragungsvektor $\underline{\tau}$ bildet sich aus den Anpresskräften \underline{F}_N und dem Reibbeiwertsvektor $\underline{\mu}$. Es soll hierbei beachtet werden, dass durch mechanische Vorspannung der Räder die Anpresskräfte einstellbar und unabhängig von der Kinematik des Systems sein können.

$$\underline{\tau} = \underline{F}_N \cdot \underline{\mu} \quad (3)$$

Dieses Modell ist offensichtlich nicht-linear und aus der Modellherleitung in [1] und [4] folgt das die Übertragungsmatrizen von Winkelfunktionen abhängen.

1.3 Linearisierung und Zustandsraum

Für die Spätere Reglersynthese ist das ableiten eines linearen Zustandsraummodells zu jedem Zeitschritt notwendig. Der Vorgang basiert auf einer Taylor-Reihenentwicklung unter Verwendung der Jacobi-Matrizen des nicht-linearen Modells. Die Taylor-Reihe führt auf die linearisierte Funktion Δy_{lin} an einem beliebigen Arbeitspunkt AP zu einem Zeitschritt i :

$$\Delta y_{lin} = f \Big|_{AP_i} + \underline{\underline{J}}_{\underline{u}}(i) \cdot \Delta \underline{u} + \underline{\underline{J}}_{\underline{q}}(i) \cdot \Delta \underline{q} + \underline{\underline{J}}_{\dot{\underline{q}}}(i) \cdot \Delta \dot{\underline{q}} + \underline{\underline{J}}_{\ddot{\underline{q}}}(i) \cdot \Delta \ddot{\underline{q}} = 0 \in \mathbb{R}^5 \quad (4)$$

Mit den partiellen Differentialen als Jacobi- Matrizen:

$$\begin{aligned} \underline{J}_{\underline{u}}(i) &= \left. \frac{\partial f}{\partial \underline{u}} \right|_{AP_i}; \underline{J}_{\underline{q}}(i) = \left. \frac{\partial f}{\partial \underline{q}} \right|_{AP_i}; \\ \underline{J}_{\underline{\dot{q}}}(i) &= \left. \frac{\partial f}{\partial \underline{\dot{q}}} \right|_{AP_i}; \underline{J}_{\underline{\ddot{q}}}(i) = \left. \frac{\partial f}{\partial \underline{\ddot{q}}} \right|_{AP_i} \end{aligned} \quad (5)$$

Das linearisierte Modell am aktuellen Arbeitspunkt ist entsprechend:

$$\Delta \underline{\ddot{q}} = \underbrace{(-\underline{J}_{\underline{\ddot{q}}}(i))^{-1}}_{\text{für}} + \underbrace{(-\underline{J}_{\underline{\dot{q}}}(i))^{-1}}_{\text{für}}$$

Woraus sich das tet:

$\underline{\dot{x}}$

\underline{y}

Das Modell besi

- $\underline{x} \in \mathbb{R}^m$ Zus
- $\underline{y} \in \mathbb{R}^p$ Aus
- $\underline{u} \in \mathbb{R}^n$ Eingangsvektor
- $\underline{z} \in \mathbb{R}^q$ Störvektor

und charakterisierenden Matrizen:

- $\underline{A} \in \mathbb{R}^{m \times m}$ Dynamikmatrix
- $\underline{B} \in \mathbb{R}^{m \times n}$ Eingangsmatrix
- $\underline{C} \in \mathbb{R}^{p \times m}$ Ausgangsmatrix
- $\underline{D} \in \mathbb{R}^{p \times n}$ Durchgriff
- $\underline{E} \in \mathbb{R}^{m \times q}$ Störmatrix

Dieses Modell ist nun mit einer beschränkten Gültigkeit eng um den gewählten (oder gemessenen) Arbeitspunkt AP_i linear.

2 Reglersynthese

Der hier vorgestellte nicht-lineare Regler in Abbildung 3 basiert auf der Methode der Online-Linearisierung, optimiert aber zudem den Stellvektor anhand einer Polvorgabe. Diese wiederum nutzt optimale Pole eines geschlossenen linearen Modellregelkreises, welcher durch eine Form der Fehlerflächenreduktion gefunden wird. Wenn der Regler diese idealen Polstellen dem Gesamtsystem in jedem Zeitschritt und an jedem beliebigen Arbeitspunkt aufzwingen kann, so ist das ge

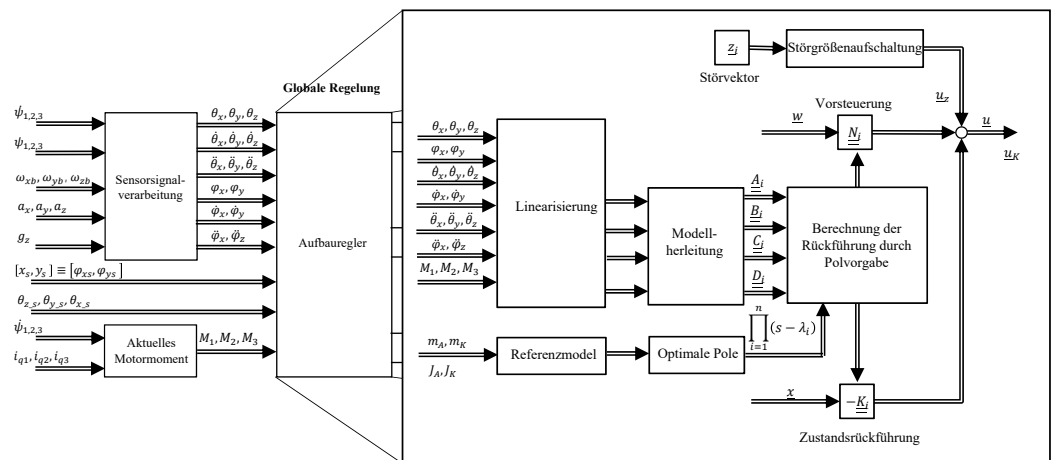


Abbildung 3: nicht-linearer Polvorgaberegler

22.02.2021

30

2.1 Referenzmodell und optimale Pole

Auf Basis des in Kapitel 1.3 vorgestellten, linearen Zustandsraummodells der Regelstrecke sollen Polstellen abgeleitet werden, die für das ideale System am Referenzarbeitspunkt der quasi-stationären Ruhelage gelten. Dafür wird zunächst ein fiktiver idealer Regelkreis angenommen. da aber nur die Polstellen von Interesse sind, ist die Berechnung einer expliziten Stellgrößenrückführung nicht notwendig. Es wird eine Fehlerfunktion anhand der bereits in [1] vorgestellten Methode der Zustandsregelung angenommen:

$$\begin{aligned} \min_u (J_{LQ}) &= \frac{1}{2} \underline{x}^T(t_e) \cdot \underline{F} \cdot \underline{x}(t_e) \\ &+ \frac{1}{2} \int_{t_0}^{t_e} \underline{x}^T(t) \cdot \underline{Q} \cdot \underline{x}(t) + \underline{u}^T(t) \cdot \underline{R} \cdot \underline{u}(t) dx \end{aligned} \quad (8)$$

Mit den Gewichtungsfaktoren Q und R . Aus dieser wird zunächst die Hamilton-Funktion $\mathcal{H}(\underline{x}, \underline{u}, \underline{\lambda})$ abgeleitet:

$$\mathcal{H}(\underline{x}, \underline{u}, \underline{\lambda}) = \frac{1}{2}(\underline{x}^T(t) \cdot \underline{Q} \cdot \underline{x}(t)) + \underline{u}^T(t) \cdot \underline{R} \cdot \underline{u}(t) + \underline{\mathcal{L}}^T(t)(\underline{A} \cdot \underline{x} + \underline{B} \cdot \underline{u}) \quad (9)$$

Aus den Optimierungsbedingungen folgt:

$$\begin{aligned} \underline{\dot{\mathcal{L}}}^T(t) &= (\underline{\dot{P}} + \underline{P} \cdot \underline{A} - \underline{P} \cdot \underline{B} \cdot \underline{R}^{-1} \cdot \underline{B}^T \cdot \underline{P}) \cdot \underline{x}(t) \\ \underline{\mathcal{L}}^T(t) &= -(\underline{A}^T \cdot \underline{P} + \underline{Q}) \cdot \underline{x}(t) \end{aligned} \quad (10)$$

Es ergibt sich die Ricatti-Gleichung mit Nebenbedingungen:

$$-\underline{\dot{P}} = \underline{A}^T \cdot \underline{P} - \underline{P} \cdot \underline{B} \cdot \underline{R}^{-1} \cdot \underline{B}^T \cdot \underline{P} + \underline{Q} + \underline{P} \cdot \underline{A} = 0 \quad (11)$$

Aus dieser lassen sich durch Lösen der Gleichung und Umformung die Polstellen leicht berechnen:

$$\det(s\underline{I} - \underline{A} + \underline{B} \cdot \underline{R}^{-1} \cdot \underline{B}^T \cdot \underline{P}) = \prod_{i=1}^n (s - \lambda_i) \quad (12)$$

Diese Polstellen repräsentieren nun das ideale geschlossenen System, die weiteren Schritte dienen dazu dem realen, geschlossenen Regelkreis dieses dynamische Verhalten aufzuzwingen.

2.2 Linearisierung und Modellherleitung

Um die Regelung an jedem beliebigen Arbeitspunkt durchführen zu können, soll auf Basis der in Kapitel 1.3 hergeleiteten Jacobimatrizen der nicht-linearen Strecke ein zu dem jeweiligen AP gültiges Ersatzmodell in Zustandsraumdarstellung hergeleitet werden.

$$\begin{aligned} \underline{\dot{x}} &= (\underline{A}_i - \underline{B}_i \cdot \underline{K}_i) \cdot \underline{x} \\ \underline{y} &= (\underline{C}_i - \underline{D}_i \cdot \underline{K}_i) \cdot \underline{x} \end{aligned} \quad (13)$$

Dieses dient der nachgelagerten Polvorgabe als Grundlage zur Berechnung des Zustandsrückführungsvektors. Da dieser Prozess in jedem Zeitschritt auf Basis der nicht-linearen Jacobimatrizen durchgeführt wird, ist er für die Echtzeitfähig kritisch. Der vorliegende Beitrag konzentriert sich auf die Optimierung dieses Prozesses auf Basis einer heuristischen Näherung der Ausgangswerte $\underline{A}_i, \underline{B}_i, \underline{C}_i, \underline{D}_i$ bei jedem Zeitschritt i .

2.3 Zustandsrückführung und Polvorgabe

Auf Basis des online-linearisierten Zustandsraummodells zum Zeitschritt i am aktuellen Arbeitspunkt und den idealen Polen p wird nun durch Polvorgabe der Zustandsrückführungsvektor \underline{K}_i und der Vorfilter \underline{N}_i berechnet. Dazu werden die geregelten Pole des Systems

$$\det(s\underline{I} - \underline{A}_i + \underline{B}_i \cdot \underline{K}_i) = \prod_{i=1}^n (s - \lambda_i) = \text{const.} \quad (14)$$

mit den optimalen Polen gleichgesetzt.

$$\det(s\underline{I} - \underline{A}_i + \underline{B}_i \cdot \underline{K}_i) = \det(s\underline{I} - \underline{A} + \underline{B} \cdot \underline{R}^{-1} \cdot \underline{B}^T \cdot \underline{P})$$

So kann durch Koeffizientenvergleich ein Zustandsrückführvektor \underline{K}_i berechnet werden. Ein Vorfilter soll das stationäre Verhalten verbessern. Die inverses der internen Systemdynamik ist allerdings aufgrund der Dimensionen nicht explizit zu berechnen und wird deshalb durch eine Pseudoinverse gebildet:

$$\underline{N}_i = -(\underline{C}(\underline{A}_i + \underline{B}_i \cdot \underline{K}_i)^{-1})^+ \quad (15)$$

Es ergibt sich so das Stellgesetz des Reglers zu:

$$\underline{u} = -\underline{K}_i \cdot \underline{x} + \underline{N}_i \cdot \underline{w} \quad (16)$$

Für der Sollwerteingang \underline{w} ist der geschlossener Regelkreis nun linear und hat stabile, konstante Pole.

3 Echtzeioptimierung der Reglerfunktionen

Der in Kapitel 2 vorgestellte Regelalgorithmus arbeitet teilweise mit den Jacobi-Matrizen der nicht-linearen Strecke, welche Winkelfunktionen enthalten und deshalb die Echtzeitausführung des Codes auf einer Zielhardware besonders beeinflussen. Die Funktionen können teilweise durch eine heuristische Näherung ersetzt werden, welche Aufgrund der noch aufzuzeigenden niedrigeren Zeitkomplexität eine Chance zur Optimierung der Echtzeitanforderung hinsichtlich Genauigkeit, Rechenarchitektur und Berechnungsschrittweite bietet. In diesem Kapitel soll nun zunächst die Berechnungsdauer durch Klassifizierung der verwendeten Algorithmen nach der erstmals von Bachmann [5] verwendeten Landau Notation [6] in Abhängigkeit der Eingänge und Rechenoperationen prädiert werden.

3.1 Herleitung der Komplexität des nicht-linearen Modells

Die Simulation anfangswertbehafteter, gewöhnlicher Differentialgleichungen führt unweigerlich auf die Verwendung numerischer Integrationsverfahren. Eine Untersuchung des optimalen Verfahrens unter Echtzeitanforderungen sollte mit einer Betrachtung des lokalen und globalen Fehlers der Integrationsmethode beginnen. Ausgehend von der Landau Notation [6] hat z.B. das einfachste Euler Verfahren 2. Ordnung einen lokalen Fehler von $\mathcal{O}(h^2)$ und einen globalen Fehler $\mathcal{O}(h)$ für die Schrittweite h . Das Heun- Trapezverfahren 3. Ordnung einen lokalen Fehler von $\mathcal{O}(h^3)$ und einen globalen Fehler $\mathcal{O}(h^2)$. Der maximale lokale Fehler eines Verfahrens der Ordnung p (für kleine Schrittweiten h) ist also eine Funktion proportional zu h^p oder mit Landau-Symbolen: $\mathcal{O}(h^p)$ [7]. Es entsteht dadurch ein Zielkonflikt, da ein höherwertiges Verfahren einen geringeren Fehler und somit eine größere Zeitschrittweite erlaubt. Es ist also nicht in allen Fällen davon auszugehen, dass ein einfaches Verfahren bei vorgegebener Fehlertoleranz grundsätzlich schneller rechnet, ein komplexeres kann diesen durch ein Aufweiten der Schrittweite kompensieren. Die Echtzeitanforderung muss also in Abhängigkeit des Fehlers und der Schrittweite optimiert werden.

Es ist weiterhin davon auszugehen, dass die dem nicht-linearen Modell zugrundeliegenden trigonometrischen Funktionen hinsichtlich ihrer Algorithmen-Komplexität die Echtzeitfähigkeit signifikant beeinflussen. Zur Berechnung der Winkelfunktionen wird von einer Darstellung über den CORDIC Algorithmus ausgegangen. Grundlage ist die Drehung eines Einheitsvektors, der in x-Achse liegt, um einen Winkel Θ .

$$\begin{bmatrix} x_n \\ y_n \end{bmatrix} = \begin{bmatrix} \cos \Theta & -\sin \Theta \\ \sin \Theta & \cos \Theta \end{bmatrix} = \cos \Theta \cdot \begin{bmatrix} 1 & -\tan \Theta \\ \tan \Theta & 1 \end{bmatrix}$$

Die Drehung wird aufgrund der Übergänge der Winkelfunktionen als Teildrehung $\sigma_i \in \{-1, 1\}$ um Teilwinkel α_i realisiert und entspricht:

$$\Theta = \sum_i \sigma_i \cdot \alpha_i \text{ mit } \sigma_i = \begin{cases} -1 & \text{falls } z_i \leq 0 \\ 1 & \text{sonst} \end{cases} \quad (17)$$

Über den Zusammenhang $\tan \alpha_i = 2^{-i}$ erhält man den

folgenden Algorithmus:

$$\begin{aligned} \begin{bmatrix} x_n \\ y_n \end{bmatrix} &= \prod_{i=0}^{n-1} \cos \alpha_i \begin{bmatrix} 1 & -\sigma_i 2^{-i} \\ \sigma_i 2^{-i} & 1 \end{bmatrix} \cdot \begin{bmatrix} x_0 \\ y_0 \end{bmatrix} \\ \begin{bmatrix} x_n \\ y_n \end{bmatrix} &= K \cdot \prod_{i=0}^{n-1} \begin{bmatrix} 1 & -\sigma_i 2^{-i} \\ \sigma_i 2^{-i} & 1 \end{bmatrix} \cdot \begin{bmatrix} x_0 \\ y_0 \end{bmatrix} \end{aligned} \quad (18)$$

mit dem Skalierungsfaktor $K = \prod_{i=0}^{n-1} \cos \alpha_i \approx 0,60725...$ für $n \rightarrow \infty$, es entsteht die folgende, einfache Formelschar:

$$\begin{aligned} x_{i+1} &:= x_i - \sigma_i 2^{-i} y_i \\ y_{i+1} &:= y_i + \sigma_i 2^{-i} x_i \\ z_{i+1} &:= z_i - \sigma_i \arctan 2^{-i} \end{aligned} \quad (19)$$

Deren Algorithmen-Komplexität über die Multiplikation und die arctan-Funktion bestimmt sind. Der naive aus der Schule Bekannte „menschliche“ Algorithmus zur Multiplikation hat die Laufzeit $\mathcal{O}(n^2)$ [8], wenn als Effizienzmaß die Bitkomplexität auf mehrbändigen Turingmaschinen, also die maximale Laufzeit des Algorithmus gemessen als benötigte Bitoperationen in Abhängigkeit von der Bitlänge n der Eingabegrößen gewählt wird. Nach [9] hat der momentan effizienteste Multiplikationsalgorithmus eine Zeitkomplexität von:

$$\mathcal{O}(n \log n) \quad (20)$$

Die Zeitkomplexität in Abhängigkeit der Genauigkeit n (Anzahl der signifikanten Bits) der arctan-Funktion lässt sich über dessen Verbindung zum komplexen Logarithmus Naturalis herleiten [10]:

$$\arctan 2^{-i} = \frac{1}{2i} \ln \frac{1 + i2^{-i}}{1 - i2^{-i}} \quad (21)$$

Die Algorithmen-Komplexität des komplexen Logarithmus ist über die Padé Approximation gegeben [11]:

$$\mathcal{O}(M(n) \log n) \quad (22)$$

Wobei $M(n)$ die bereits vorgestellte Komplexität des eingesetzten Multiplikationsalgorithmus ist.

3.2 Herleitung der Komplexität des Neuronalen Netzes

Das heuristisch genäherte Modelle hingegen wird als gerichteter Graph $G = f(U, C)$ aufgefasst, dessen Knoten $j \in U$ die Neuronen und dessen Kanten $c \in C$ die

Neuronenverbindungen sind. Jede Verbindung $(i, j) \in C$ innerhalb einer Neuronenschicht k besitzt eine Gewichtung w_{ij} . Die Netztopologie einer Neuronenschicht kann in Form einer $(d \times d) | d = |U|$ dimensionalen Adjazenzmatrix $\underline{A} = (a_{ij}^k)$ mit den Elementen:

$$a_{ij}^k = \begin{cases} 1 & (i, j) \in C \\ 0 & (i, j) \notin C \end{cases} \quad (23)$$

beschrieben werden. Sie bestehen aus topologisch zusammenhängenden Neuronen welche immer auf der Basis eines allgemeinen Neurons ihre gewichteten Eingänge x_1, \dots, x_n mit den Gewichtungsfaktoren w_{1j}, \dots, w_{nj} zunächst mit der folgenden Übertragungsfunktion/Integrationsfunktion (Summation) in eine Netzausgabe wandeln:

$$net_j^k = \sum_{i=1}^{m^{k-1}} j_i^{k-1} = \sum_{i=1}^{m^{k-1}} w_{ij}^{k-1} \cdot out_i^{k-1} \quad (24)$$

Das Ergebnis durchläuft eine Aktivierungsfunktion act_j^k welche über einen erlernten oder vorgegebenen Schwellenwert θ_j die Ausgabefunktion z.B. die Sigmoide aktiviert:

$$out_j^k = f_j(act_j^k) \text{ mit } act_j^k = \begin{cases} net_j^k & net_j^k > \theta_j \\ 0 & net_j^k \leq \theta_j \end{cases} \quad (25)$$

Es ergibt sich so für eine Netzschicht k das folgende Gleichungssystem:

$$\begin{aligned} net^k &= \underline{w}^{k-1} \cdot out^{k-1} \\ out^k &= f(net^k) \end{aligned} \quad (26)$$

mit den Dimensionen

$$\begin{aligned} net^k &\in \mathbb{R}^{n^k \times 1} \\ \underline{w}^{k-1} &\in \mathbb{R}^{n^k \times n^{k-1}} \\ out^{k-1} &\in \mathbb{R}^{n^{k-1} \times 1} \end{aligned} \quad (27)$$

Es ist also ersichtlich das zur Lösung einer Netzschicht k mit einer einfachen Aktivierungsfunktion der Komplexität $\mathcal{O}_{trans}^{(k-1) \rightarrow k} = \mathcal{O}(n^k)$ grundsätzlich nur eine dreidimensionale Matrizenmultiplikation durchgeführt werden muss. diese hat eine Zeitkomplexität von $\mathcal{O}_{mult} = \mathcal{O}(a \cdot b \cdot c)$ mit den diskreten Dimensionsachsen

a, b, c der Matrize. So lässt sich also ableiten:

$$\mathcal{O}(\underbrace{(n^k \cdot n^{k-1} \cdot 1)}_{\mathcal{O}_{mult}} + \underbrace{(n^k)}_{\mathcal{O}_{trans}}) \quad (28)$$

Für ein einfaches Neuronales Netz mit einer Eingangsschicht $k = 0$, einer Ausgangsschicht $k = 2$ und einem dazwischenliegenden sog. Hidden Layer $k = 1$ ergibt sich exemplarisch also folgende Zeitkomplexität:

$$out^2 = out^1 + out^0 = \mathcal{O}((n^2 \cdot n^1) + (n^2) + (n^1 \cdot n^0) + (n^1))$$

Unter der Annahme das die Aktivierungsfunktion für alle Layer gleich ist, ist die Komplexität des einfachen Netzes:

$$\mathcal{O}((n^0 + n^2) \cdot 2n^1) \quad (29)$$

Allgemein für ein beliebig großes Netz folgt so:

$$\begin{aligned} \mathcal{O}_{trans} &= \sum_{i=1}^k n^i \\ \mathcal{O}_{mult} &= \sum_{i=1}^k (n^{i-1} \cdot n^i) \\ &= \underbrace{\sum_{j=1}^{i-1} (n^{j-1} \cdot n^j)}_{\text{const}} + \underbrace{(n^{i-1} \cdot n^{i+1})n^i}_{\text{linear: } \mathcal{O}(n^i)} + \underbrace{\sum_{j=i+2}^k (n^{j-1} \cdot n^j)}_{\text{const}} \end{aligned} \quad (30)$$

Es ist zu erkennen das die Komplexität des Netzes von der Anzahl der in einem Layer enthaltenen Neuronen abhängt, da eine Erhöhung dieser zu einem linearen Anstieg der Berechnungskomplexität führt.

3.3 Algorithmus zur Auslegung der heuristischen Funktion

Auf Basis der gewonnenen Erkenntnisse lässt sich zunächst zusammenfassen, dass ein zu großes Netz eine höhere Zeitkomplexität als die Ursprungsfunktionen aufweist, ein zu kleines Netz oder aber zu einfache Aktivierungsfunktionen u.U. die geforderte Genauigkeit nicht erreichen. Es entsteht ein Zielkonflikt, den es über eine Optimierung zu lösen gilt. Dazu ist es zunächst notwendig die Struktur der zu ersetzenden nicht-linearitäten zu untersuchen. Dazu kann der Zusammenhang zwischen den linearisierten Teilmodellen in Zustandsraumdarstellung zu dem Referenzmodell betrachtet werden. Hierbei zeigen sich Zellen innerhalb der Matrizen $\underline{A}_i, \underline{B}_i$ die sich aufgrund linearer Zusammenhänge bei unterschiedlichen Arbeitspunkten nicht än-

Konzept: Evolutionärer Algorithmus zur Optimierung Neuronaler Netze unter Berücksichtigung der Komplexität

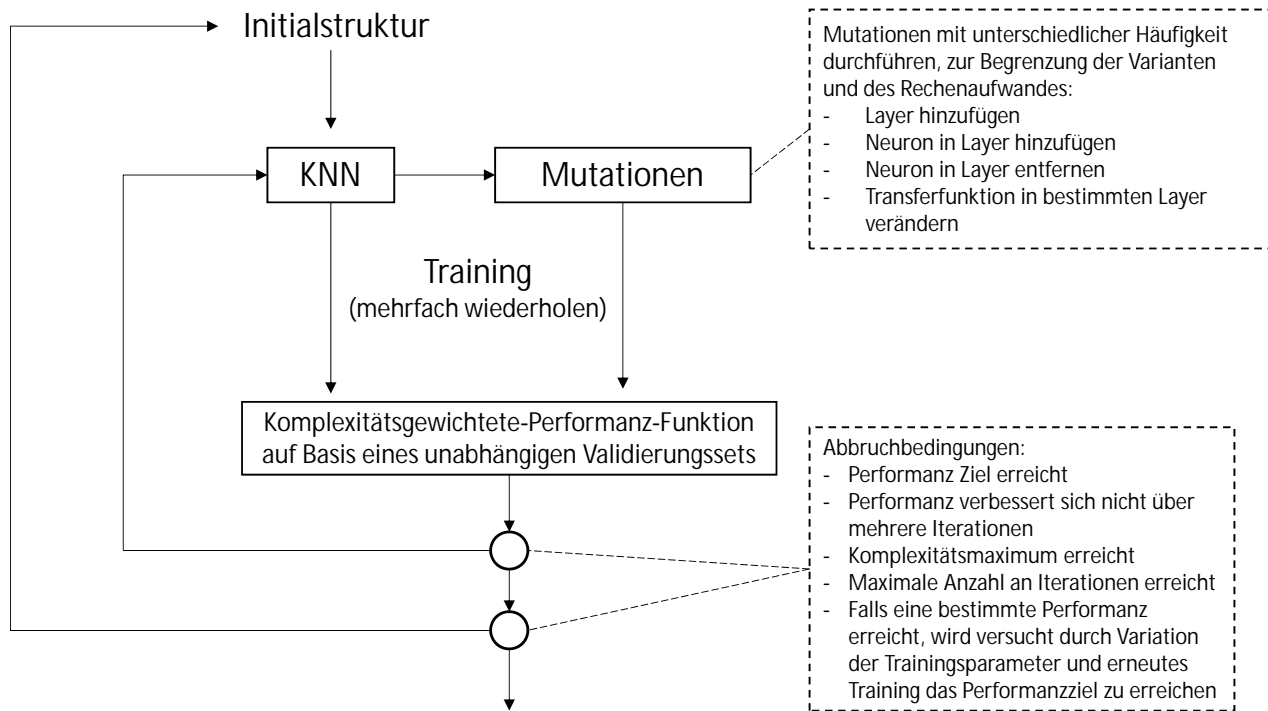


Abbildung 4: Optimierungsalgorithmus zum finden eines heuristischen Ersatzmodells

dern. Diese Zellen können als ideales lineares Übertragungsverhalten separiert werden. Die reduzierten Elemente sollen durch ein heuristisches Modell (hier: neuronales Netz) genähert werden. Dazu wird ein Trainingsalgorithmus entworfen der unter Beachtung der Algorithmen-Komplexität sowohl der Netzstruktur als auch der Aktivierungsfunktionen des Netzes eine optimale Lösung findet. Die folgende Abbildung 4 zeigt den Ansatz.

Zunächst müssen die Matrizenelemente so normiert werden, dass sie zu dem Wertebereich der Aktivierungsfunktion passen. es wird eine Normierung sowohl für die Netzeingänge als auch die Trainingsdaten auf einen Wertebereich von $[-1 : 1]$ durchgeführt. Nach dem Training wird die Normierung invers als Bias in die Ausgangsschicht zurückgespielt. Eine beliebige Initialstruktur wird mittels Mutation (siehe dazu auch [12]) hinsichtlich ihrer Neuronenanzahl, Schichttiefe, und Aktivierungsfunktion im jeweiligen Layer so lange trainiert und mit den vorherigen Netzen verglichen, bis sich ein Optimum einer Kostenfunktion ergibt. Die Kostenfunktion C setzt sich aus der Fehleranfälligkeit C_E und der Algorithmen-Komplexität C_O zusammen:

$$C = C_E \cdot C_O \quad (31)$$

Die Algorithmen-Komplexität C_O wurde bereits im vorherigen Kapitel ausführlich besprochen und setzt sich wie folgt zusammen:

$$C_O = \sum (\alpha_{g(x)} \cdot n_{g(x)}) + n_{mult} \quad (32)$$

Sie wird aus der Anzahl der Produktsummen n_{mult} , der Anzahl der Berechnungen, die für das Ausführen einer Aktivierungsfunktion nötig sind, $n_{g(x)}$ und des auf die Produktsumme normierten Gewichtungsfaktors der Aktivierungsfunktion $\alpha_{g(x)}$ berechnet. Dabei ist zu beachten das für Ausführen einer Aktivierungsfunktion hier eine von der Anzahl der Neuronen $n^{(i)}$ in einem Layer i abhängige Funktion angenommen wird, die über alle Layer k hinweg summiert wird:

$$n_{g(x)} = \sum_{i=1}^k (n^{(i)}) \quad (33)$$

Die Anzahl der Produktsummen n_{mult} muss bedingt durch die Struktur des Netzes in gleicher Weise aufsummiert werden:

$$n_{mult} = \sum_{i=1}^k (n^{(i-1)} \cdot n^i) \quad (34)$$

Die Fehleranfälligkeit C_E ist hingegen von den Ausgabewerten des Netzen $y_{i,net}$ in Relation zu den in den Trainingsdaten gespeicherten Ergebnissen $y_{i,train}$ abhängig und wird über die Anzahl der auswertbaren Daten d aufsummiert:

$$C_E = \frac{1}{d} \sum_{i=1}^d (y_{i,train} - y_{i,net})^2 \quad (35)$$

Der Fehler des mutierten Netzes muss sich somit überproportional verringern gegenüber der Komplexität, damit das Netz gegenüber dem Initialnetz bevorzugt wird. Somit wird das Netz zu einer besseren Echtzeitfähigkeit hin optimiert, ohne den Fehler zu verschlechtern.

4 Zusammenfassung

Im vorliegenden Beitrag wurde auf Grundlage eines nicht-linearen Modells der Regelstrecke eines Intralogistik- Transportsystems „S-Mobile“ zunächst ein optimierender, nicht-linearen Polvorgaberegler auf Basis der Jacobi-Matrizen entwickelt. Bedingt durch die nicht-linearitäten, vor allem der trigonometrischen Funktionen innerhalb des nicht-linearen Streckenmodells und damit auch in den Jacobi-Matrizen, wurde der Einfluss auf die Echtzeitfähigkeit untersucht. Mit der Landau Notation der Algorithmen-Komplexität der zugrundeliegenden Berechnungsalgorithmen der Grundfunktionen, wurde eine Methode zur Prädiktion der zu erwartenden Berechnungszeit vorgestellt. Auf dieser Grundlage wurde ein Algorithmus entwickelt, welcher ein heuristisches Modell als Surrogate-Modell der nicht-Linearitäten so trainiert, dass sowohl die Abweichung zu der Referenz klein als auch die Algorithmen-Komplexität insgesamt geringer ist.

Danksagung

Der vorliegende Beitrag wurde im Rahmen des Teilprojektes auto-EVM als Teil des Verbundprojekts autoMoVe unter dem Förderkennzeichen ZW6-85030889 gefördert. Die Verantwortung für den Inhalt liegt bei den Autoren. Für die Förderung bedanken sich diese herzlichst.



Referenzen

- [1] Liu-Henke X, Göllner M, Tao H. An intelligent control structure for highly dynamic driving of a spherical electrical drive. In: *2017 Twelfth International Conference on Ecological Vehicles and Renewable Energies (EVER)*. Piscataway, NJ: IEEE. 2017; pp. 1–10.
- [2] Göllner M, Zhang J, Liu-Henke X. Model predictive trajectory control for automated driving of a spherical electrical drive. In: *2018 IEEE International Systems Engineering Symposium (ISSE)*. IEEE. 10/1/2018 - 10/3/2018; pp. 1–6.
- [3] Liu-Henke X, Göllner M, Tao H. Modellbasierte Reglerauslegung eines sphärischen Elektroantriebs. In: *Workshop der ASIM/GI-Fachgruppen "Simulation technischer Systeme", "Grundlagen und Methoden in Modellbildung und Simulation"*, edited by Wahmkow C, Roßmanek P, Wendorf R, ASIM-Mitteilungen, pp. 37–46. Stralsund: Fachhochschule Stralsund. 2015;.
- [4] Göllner M, Liu-Henke X, Frerichs L. Analyse und Simulation des Kraftübertragungsverhaltens von Mecanum-Rädern. In: *Proceedings ASIM SST 2020*, edited by Deatcu C, Lückerrath D, Ullrich O, Durak U, ASIM Mitteilung, pp. 89–98. Wien: ARGESIM. 2020;.
- [5] Bachmann PGH. *Die analytische Zahlentheorie*, vol. 2 of *Zahlentheorie*. Leipzig: B.G. Teubner. 1894.
- [6] Landau E. *Handbuch Der Lehre Von Der Verteilung Der Primzahlen*. Leipzig and Berlin: B.G. Teubner. 1909.
- [7] Berg I. A Comparison of Numeric Integration Schemes: Runge Kutta vs. Euler: On the Influence of time step sizes on the accuracy of numerical simulations.
- [8] Newman JR. *The World of Mathematics*, vol. 1. New York, NY: Simon and Schuster. 1956.
- [9] Harvey D, van der Hoeven J. Integer multiplication in time $O(n \log n)$. *Princeton University, Department of Mathematics*. 2019;.
- [10] Cotes R. Logometria. *Philosophical Transactions Royal Society*. 1714;.
- [11] Padé H. Sur la généralisation des fractions continues algébriques. *Journal de Mathématiques Pures et Appliquées*. 1894;pp. 291–330. URL <http://eudml.org/doc/235790>
- [12] Yarom OA, Jacobitz S, Liu-Henke X. Design of Genetic Algorithms for the Simulation-Based Training of Artificial Neural Networks in the Context of Automated Vehicle Guidance. In: *19th International Conference on Mechatronics – Mechatronika*, edited by Maga D. 2020; pp. 254–261.

Simulation Study on Methods for Modeling Elastic Collisions

Milena Sipovac^{1*}, Andreas Körner¹

¹Institute of Analysis and Scientific Computing, TU Wien, Wiedner Hauptstraße 8-10, 1040 Vienna, Austria; *milena.sipovac@tuwien.ac.at

Introduction

This paper is focused on investigating and describing collision processes and comparing the numerical results of different ODE solvers. The collisions are modeled on a system of two points of mass moving along circular paths. The equations of motion are derived using the Euler-Lagrange method. All the simulations are done in MATLAB, and the derivations of the equations are done in Maple. Explicit Euler and Heun methods are implemented, and the results of both solvers were compared to the ode45 solver of MATLAB, which is used as reference due to complexity of the equations of movement of this coupled system.

The main question that this work aims to answer is how the different algorithms for solving differential equations behave when altered in a way that the collisions can be modeled.

The idea of this work is based on the problem of detecting neighborhoods of two objects while a movement occurs. As seen in the Figure 1, we observe mass points m_1 and m_2 moving along a certain path. Both of these points have fixed circular neighbourhoods U_1 and U_2 , and cannot find each other in their respective neighbourhoods. When the distance between these two points is smaller than the radius of any of the circles, a certain type of collision happens. We will simplify this generalized idea in the next chapter. The distance between two points needs to be calculated at all times in this model, and has to be stopped when two objects find themselves within a given radius of each other. This was implemented in two different differential equation solvers, explicit Euler and Heun.

1 Description of the System

The system observed is a system of two mass points m_1 and m_2 moving along a circular path in a two-dimensional space. This can be seen in the Figure 2.

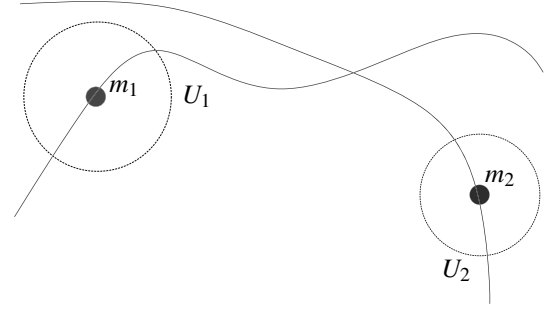


Figure 1: A sketch of the generalised idea of this work. The points (blue and red) are moving along their paths, and have masses m_1 and m_2 , with the respective neighbourhoods $U_{1,2}$.

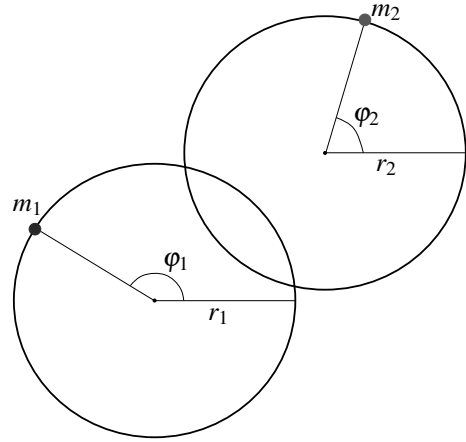


Figure 2: A sketch of the system setup, the blue dot represents the point mass m_1 and the red dot the point mass m_2 . Movement occurs in circular paths, with position vectors φ_1 and φ_2 , both with fixed radius $r_{1,2}$.

Each of them has a fixed radius of movement and can bump into each other at times.

The equations of motion are derived using the Euler-Lagrange method. The kinetic energy T_i of each point mass is

$$T_i = \frac{1}{2} m_i \dot{r}_i(t), \quad i = 1, 2. \quad (1)$$

The variable $r_i(t)$ represents the position vector of the respective points of mass. Since we are observing a circular motion, the position can be restricted to circular motion, with the displacement of the two origins of the circles by a vector $(d_x, d_y)^T$, so the constraints do not have to be additionally considered,

$$r_1(t) = \begin{pmatrix} r_1 \cos \phi_1 \\ r_1 \sin \phi_1 \end{pmatrix} \quad \text{and} \quad r_2(t) = \begin{pmatrix} r_2 \cos \phi_2 + d_x \\ r_2 \sin \phi_2 + d_y \end{pmatrix}.$$

The points are attracted to each other by gravity force so the potential energy is observed as

$$U = -\frac{Gm_1m_2}{d}, \quad (2)$$

with $d(t) = \|r_1(t) - r_2(t)\|$ representing the euclidean distance between the two points, and $G = 6,7 \cdot 10^{-11} \text{ m}^3 \text{ kg}^{-1} \text{ s}^{-2}$. The equations of motion are obtained by using the Euler-Lagrange Equation

$$\frac{\partial L}{\partial \phi_i} - \frac{d}{dt} \left(\frac{\partial L}{\partial \dot{\phi}_i} \right) = 0. \quad (3)$$

The initial conditions are set by altering the position angles ϕ_1 and ϕ_2 , as well as the angular velocities $\dot{\phi}_1$ and $\dot{\phi}_2$, which are both time-dependent.

2 Collisions

The model calculates the distance between the two points at all times and when they find each other in the defined neighborhood of each other, a collision is modeled. The reason why the distance for impact to happen was not set to zero is that the potential energy used in this model is the gravitational energy, see (2), where we divide with the distance. In order to avoid division with zero, we set the distance for impact to occur to be $d_{min} = 0.2\text{m}$. The collisions are modeled as elastic collisions, and the initial conditions are updated after each impact based on the Law of conservation of momentum and energy, as can be seen in (4).

$$\begin{aligned} m_1 \dot{\phi}_1(t^-) + m_2 \dot{\phi}_2(t^-) &= m_1 \dot{\phi}_1(t^+) + m_2 \dot{\phi}_2(t^+) \\ m_1 \dot{\phi}_1(t^-)^2 + m_2 \dot{\phi}_2(t^-)^2 &= m_1 \dot{\phi}_1(t^+)^2 + m_2 \dot{\phi}_2(t^+)^2 \end{aligned} \quad (4)$$

3 Simulation Studies

The step size of the solving methods is varied and the quality of the solutions is compared to the results of ode45 from MATLAB. Without initial velocity, the simulations take some time before a collision occurs. When an initial velocity is introduced properly, the collisions happen more often, and this is the case we are also going to observe.

The parameters are $m_1 = 10^6 \text{ kg}$, $m_2 = 10^3 \text{ kg}$, the radii of both paths are $r_1 = r_2 = 1 \text{ m}$. Since we are dealing with the gravitational energy as a potential energy, the two points need to have a large mass in order for any movement to occur, but using large numbers is prone to numeric errors, so the model was built in 10^6 scale.

Before describing the results from the explicit solvers, we will present the results obtained with MATLAB solver ode45. Here we can see that we are expecting a sinusoidal movement for the second point of mass, m_2 , which is moving around the much heavier point m_1 , which has a very restricted angular movement.

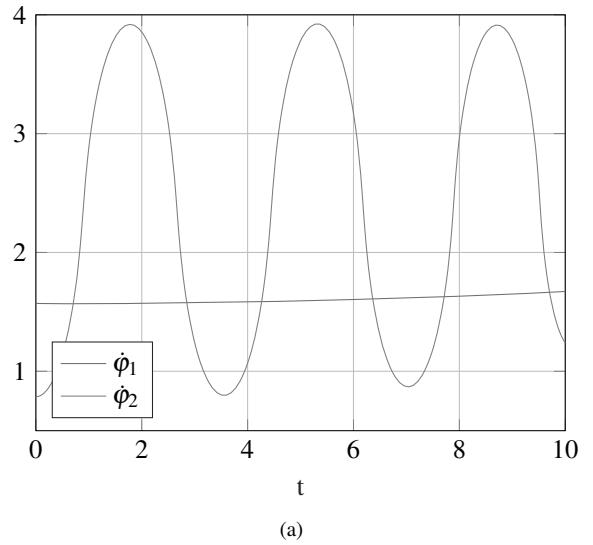


Figure 3: Simulation with MATLAB ode45, time plot of angular position, simulation time 10s.

First, we observed the movement of the particles with no initial velocity given. The initial conditions are set to $\phi_{1,0} = \frac{\pi}{6}$ and $\phi_{2,0} = \frac{\pi}{4}$, $\dot{\phi}_{1,0} = 0$ and $\dot{\phi}_{2,0} = 0$. The large difference between the two masses is necessary, otherwise the movement does not happen with no initial velocity. In these simulations, no collisions happened which was useful for heuristic comparison of

the methods. The explicit Euler method showed some stable movement at step size $h = 0.001$ and the Heun method at step size $h = 0.01$. These results can be observed in the Figure 4(a) for Euler and Figure 4(b) for Heun method.

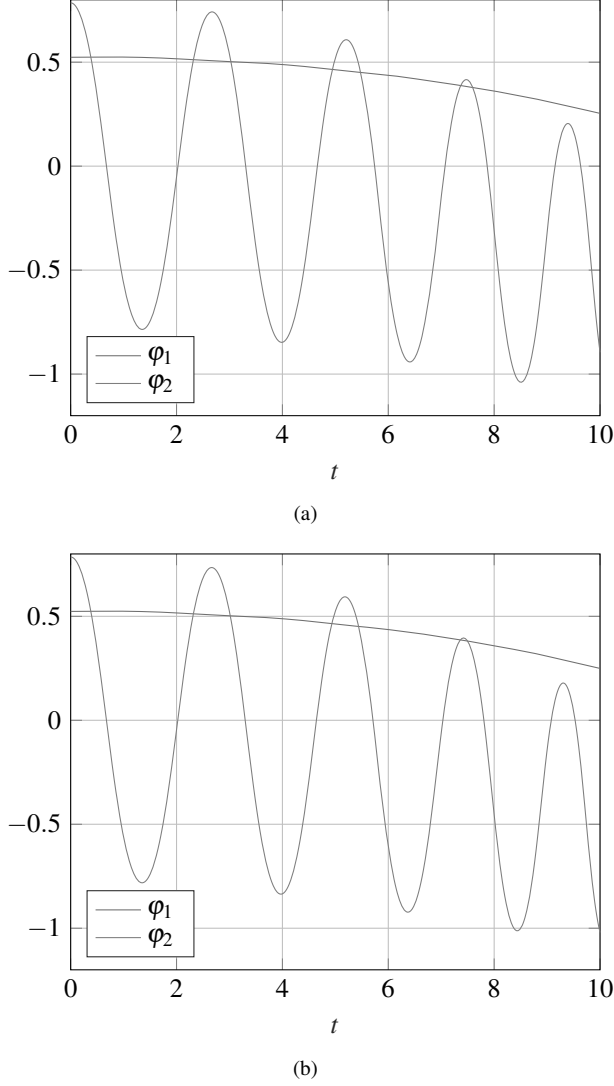


Figure 4: (a) Simulation with explicit Euler method, with step size $h = 0.001$, (b) Simulation with Heun method, with step size $h = 0.01$. Time plot of angular position. Both with simulation time 10s.

This result was used to determine the methods that will be compared when the collisions are introduced to the system. As we can see, when using the Heun method, which is also an explicit method of solving, we can observe a similar solution to the Euler method, but

with less steps and lower computational time.

After this, we introduce the initial velocity which improves the chances of a collision happening. The initial conditions are set to $\varphi_{1,0} = \varphi_{2,0} = \frac{\pi}{4}$, $\dot{\varphi}_{1,0} = 2s^{-1}$ and $\dot{\varphi}_{2,0} = 0$. Other parameters remain the same as in the previous section. We are modeling the elastic collision, and therefore expect the point with a lower mass, in this case m_2 to move in the opposite direction after the impact, which can usually be observed in a form of a jump on the time-velocity plot.

If the step size is large, when using the explicit Euler method, no collisions are detected, with the initial conditions mentioned below, as can be seen in figure 5.

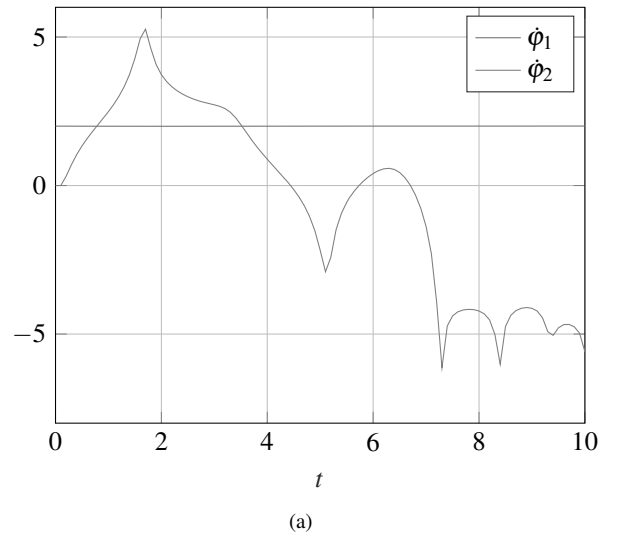
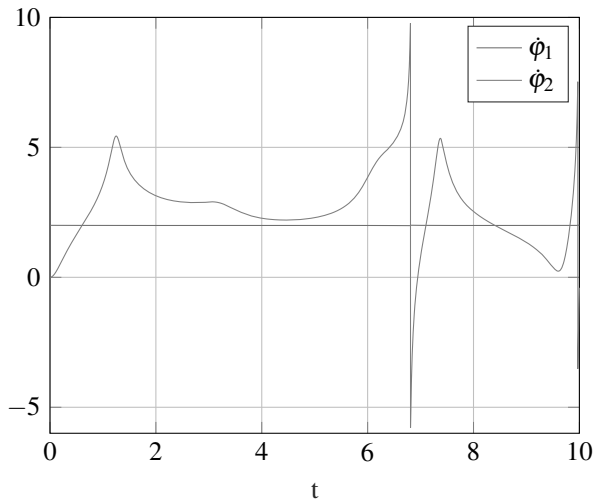


Figure 5: Simulation with explicit Euler method, with step size $h = 0.1$, for the simulation time of 10s. Time-plot of angular velocity.

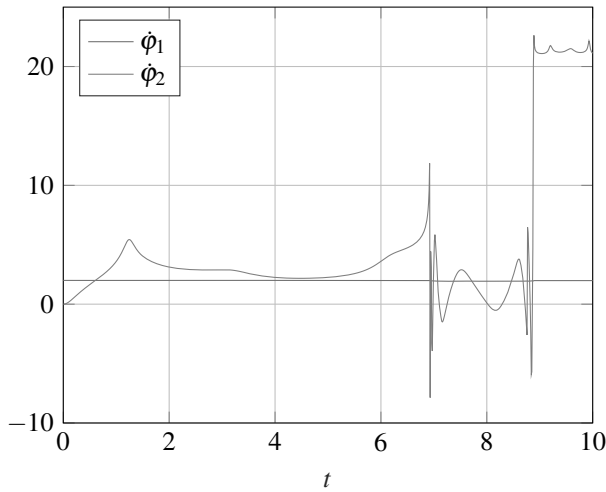
In the Figure 6 we can observe the motion with a finer step size. A collision is detected in this case, at $t = 6.809s$ and $t = 9.970s$. Since the step size is finer, the solver is more sensitive to the neighborhood radius $d_{min} = 0,2m$, and can detect the collisions more precisely.

When using Heun method with step size $h = 0.01$, collisions are detected at $t = 6.920s$ and at $t = 8.760s$.

The difference between the two methods increases after each collision, which is obvious for the mass point with lower mass, since it is exposed to more movement. The mass point with larger mass does not change its path a lot after collisions. This error can be repaired by decreasing the step size.



(a)



(b)

Figure 6: (a) Simulation with explicit Euler method, with step size $h = 0.001$. Time-plot of angular velocity. (b) Simulation with Heun method, with step size $h = 0.01$. Time-plot of angular velocity. Both for the simulation time of 10s.

4 Conclusion

This work was focused on comparing two explicit methods for solving ordinary differential equations, both modified to calculate the distance between two moving objects and stop the calculation if they find themselves within a given radius of each other. The system observed in this case were two points of mass in a two dimensional space, attracted to each other by gravity. The

equations of motion resulted in a system of two non-linear ordinary differential equations of second order. They were numerically solved by using explicit Euler and Heun methods.

In order to observe solutions, we simulated the motion without collisions by using ode45 from MATLAB. This served as a reference simulation for the rest of the work. In the next step of the computation, we simulated the motion with explicit Euler and Heun, where we varied the step size in order to determine the best simulations. As expected, the simulations both exhibited better results with a finer step size. Stable solutions were obtained for explicit Euler method for step size $h = 0.001$ and the Heun method gave similar results at step size $h = 0.01$. Instabilities and unusual behavior occurred after some of the collisions happened, which can be observed on the figures.

The equations of motion for this system are quite complicated and an analytic solution, if it exists, is out of scopes of this work. We only used numerical data for comparison. The lack of an analytic solution made it difficult to compare the results, since there was no reference, and we had to settle with choosing a good solution and doing a comparison. This could be improved by altering the system parameters so that the equations of motion have a lower complexity.

References

- [1] Witkin A. *Physically based modeling particle system dynamics*. ACM SIG- GRAPH Course Notes, 2001
- [2] Witkin A., *Physically based modeling: principles and practice constrained dynamics*. ACM SIG- GRAPH Course Notes, 2001
- [3] Hasselblatt B., Katok A. *Introduction to the Modern Theory of Dynamical Systems*. Cambridge University Press, 1995.
- [4] Hockney R. W., Eastwood J.W. *Computer simulation using particles*. crc Press, 1988.
- [5] Dudtschenko K., Küpper T., Hosham A.. The dynamics of bells as an impacting system *Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science*. 2010; 225(10):2436–2443.
- [6] Hildebrandt S., Giaquinta M.. *Calculus of Variations Volume 1 Springer-Verlag Berlin Heidelberg*. 1996.
- [7] Schek F., *Mechanics - From Newton's Laws to Deterministic Chaos Springer-Verlag Berlin Heidelberg*. 2010.

Simulationsmodell zur Generierung von Trainingsdaten für ML-gestützte Ansätze der Zustandsüberwachung

Dennis Jess^{1*}, Martin Strube^{1*}

¹ Ostfalia Hochschule für angewandte Wissenschaften, Salzdahlumer Str. 46/48, 38302 Wolfenbüttel;
[*de.jess@ostfalia.de](mailto:de.jess@ostfalia.de), m.strube@ostfalia.de

Abstract. Sowohl in der Industrie als auch im Bereich der Lehre des Maschinenbaus an Hochschulen und Universitäten lassen sich innovative ML-Projekte nur schwer umsetzen, da meistens keine geeigneten Daten für das Training von ML-Modellen zur Verfügung stehen und die Gewinnung von geeigneten Daten mit einem sehr hohen Aufwand verbunden ist. Dies gilt insbesondere für Anwendungen im Bereich der Zustandsüberwachung oder Predictive Maintenance industrieller Maschinen und Anlagen, da ML-Modelle in dieser Domäne häufig Daten vom Normalzustand und von Fehlerzuständen einer Maschine oder Anlage erfordern und historische Messdaten aus dem Betrieb industrieller Anlagen nur selten im Zusammenhang mit dokumentierten Instandhaltungsarbeiten zur Verfügung stehen. Im Rahmen dieses Beitrags wird ein Simulationsmodell eines mechatronischen Systems vorgestellt, welches die Möglichkeit zur Simulation von Fehlerzuständen und den dabei entstehenden Maschinendaten bietet. Dadurch soll der Einstieg in ML-Anwendungen für die Zustandsüberwachung von Maschinen für Studierende der Ingenieurwissenschaften und industrieller Anwender erleichtert werden.

Einleitung

Eine der größten Herausforderungen bei vielen Machine Learning (ML) Anwendungen liegt in der Verfügbarkeit geeigneter Daten für das Training von ML-Modellen [1], [2]. Dies gilt insbesondere für Anwendungen im Bereich der Zustandsüberwachung oder Predictive Maintenance industrieller Anlagen. Häufig stehen Unternehmen vor dem Problem, dass ihre Datenbestände für die Umsetzung von innovativen ML-Projekten nicht ausreichend groß sind, die vorliegenden Daten nicht den notwendigen Anforderungen entsprechen oder die erzeugten datengetriebenen ML-Modelle nicht ausreichend getestet werden können [3]. ML-Modelle für die Zustandsüberwachung von Maschinen und Anlagen erfordern häufig Daten vom

Normalzustand und von Fehlerzuständen einer Maschine oder Anlage [4]. Historische Messdaten aus dem Betrieb industrieller Anlagen stehen nur selten im Zusammenhang mit dokumentierten Instandhaltungsarbeiten zur Verfügung. Eine experimentelle Gewinnung von Trainingsdaten an Prüfständen und Maschinen ist mit einem sehr hohen wirtschaftlichen und technischen Aufwand verbunden [5], da die Erzeugung von Fehlerzuständen, die zum Beispiel durch Verschleißvorgänge an realen Maschinen und Anlagen verursacht werden, sehr zeit- und kostenintensiv ist. Dies stellt heute ein wesentliches Hemmnis für die Anwendung von ML-Verfahren für die frühzeitige Erkennung von Fehlerzuständen und Verschleißvorgängen in Maschinen und Anlagen dar [6].

Aufgrund der zuvor beschriebenen Problematik scheitern heute viele ML-Ansätze in dieser Domäne. Eine recht ähnliche Herausforderung stellt sich im Bereich der Lehre an Hochschulen und Universitäten. Durch den zunehmenden Stellenwert der Digitalisierung finden sich auch zunehmend Inhalte aus dem Themengebiet des Machine Learning auch im Studiengang Maschinenbau wieder. Ein typisches Anwendungsgebiet ist hierbei die Zustandsüberwachung von Maschinen und Anlagen. In der Lehre wird häufig auf öffentlich zugängliche Datensätze zurückgegriffen, zu denen jedoch meist nur rudimentäre Informationen zu den Spezifikationen der Maschinen und Betriebsparameter zur Verfügung stehen. Dies erschwert den Zugang für Studierende, da Ursache und Wirkung aus den Datensätzen oft nicht hervorgehen. Eine Möglichkeit, um den zuvor beschriebenen Problemen entgegenzuwirken, besteht darin, die benötigten Trainingsdaten für ML-Ansätze mithilfe eines digitalen Zwillings zu generieren. Anhand solcher Simulationsmodelle können bestimmte Fehlerbilder nachgestellt und die daraus resultierenden Messdaten künstlich erzeugt werden. Dies ermöglicht eine ingenieurwissenschaftliche Herangehensweise an die Implementierung von ML-Ansätzen.

Im Rahmen dieses Beitrags wird hierzu ein Simulationsmodell eines mechatronischen Systems vorgestellt, welches die Möglichkeit zur Simulation von Fehlerzuständen und den dabei entstehenden Maschinendaten bietet. Auf dieser Grundlage soll der Einstieg in ML-Anwendungen für die Zustandsüberwachung von Maschinen für Studierende der Ingenieurwissenschaften und industrieller Anwender erleichtert werden.

1 Auswahl eines mechatronischen Systems

Im Rahmen eines Industrie-Kooperations-Projekts zwischen der Ostfalia Hochschule für angewandte Wissenschaften und einem Industrie-Unternehmen sollten verschiedene Predictiv Maintenance-Ansätze für die Zustandsüberwachung von Baugruppen einer industriellen Lüftungsanlage untersucht werden. Das Projekt-Ziel bestand darin, die Eignung von Predictiv Maintenance-Ansätzen für die frühzeitige Erkennung von Fehlerzuständen zu bewerten und den geeignetsten Ansatz zu ermitteln. Dabei hat sich herausgestellt, dass industrielle Lüftungsanlagen eine Vielzahl von verschiedenen Fehlerbildern aufweisen, die relativ häufig auftreten und hohe Stillstands-Kosten verursachen. Zudem werden diese in der Industrie nicht nur häufig eingesetzt, sondern ähneln in den auftretenden Fehlerbildern vielen bauähnlichen Maschinen. Aufgrund der zuvor genannten Aspekte fällt die Auswahl für das mechatronische System in der Simulation auf Baugruppen einer industriellen Lüftungsanlage. In Abbildung 1 ist der Aufbau der ausgewählten Lüftungsanlage zu sehen. Die Lüftungsanlage besteht im Wesentlichen aus einem Frequenzumrichter, Asynchronmotor, Riementrieb und Ventilator.

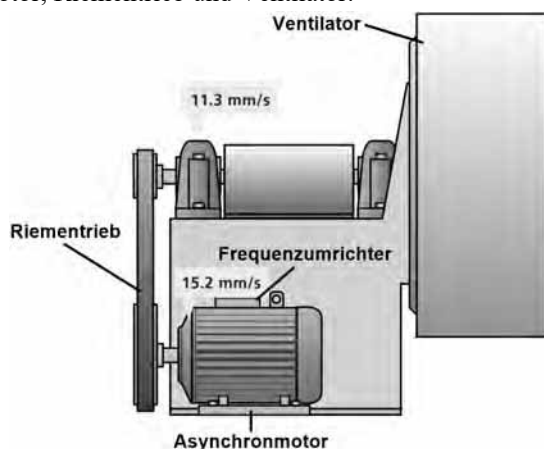


Abbildung 1: Aufbau einer Lüftungsanlage [7]

2 Modellierungsansatz

Die Erstellung des Simulationsmodells zur Lüftungsanlage erfolgt mithilfe der Software Matlab/Simulink, da diese Software vielfältige Möglichkeiten und eine umfassende Auswahl an Modell- und Softwarebibliotheken für die rechnergestützte Erstellung, Modellierung und Simulation von mathematisch darstellbaren technischen, physikalischen oder theoretischen Systemen bietet [8]. Für die Erstellung des Simulationsmodells zur Lüftungsanlage wird hauptsächlich die Simscape-Bibliothek genutzt. Diese beinhaltet viele vorgefertigte Modelle zu verschiedenen physikalischen Systemen. In Abbildung 2 ist die Struktur des Simulationsmodells zur Lüftungsanlage dargestellt. Das Simulationsmodell zur Lüftungsanlage setzt sich hauptsächlich aus einem Frequenzumrichter-, Asynchronmotor-, Riementrieb- und Ventilator-Modell zusammen.

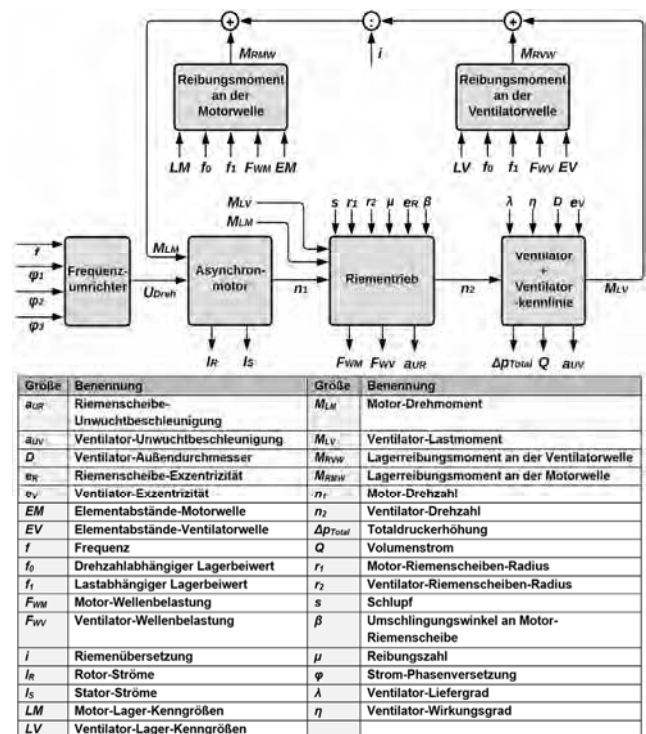


Abbildung 2: Struktur des Simulationsmodells der Lüftungsanlage

Da der Asynchronmotor von der Lüftungsanlage unterschiedliche Lastmomente aufnehmen und unterschiedliche Drehzahlen annehmen soll, wird der Asynchronmotor-Block durch einen Frequenzumrichter-Block angesteuert. Der Frequenzumrichter-Block erhält am Eingang eine Frequenz f und die drei Strom-Phasenversetzungen φ_1 , φ_2 und φ_3 . Durch die Erhöhung der Frequenz f wird die Drehmoment-Drehzahl-Kennlinie des Asynchron-

tors nach rechts verschoben. Das Prinzip zur Verschiebung der Drehmoment-Drehzahl-Kennlinie ist in Abbildung 3 dargestellt.

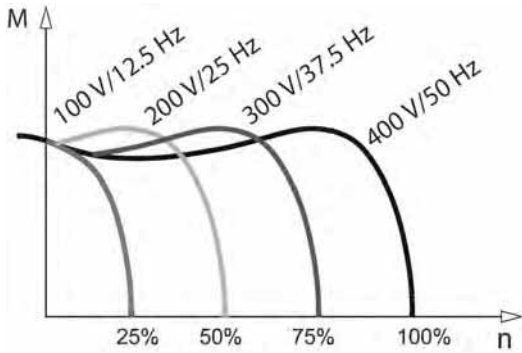


Abbildung 3: Verschiebung der Drehmoment-Drehzahl-Kennlinie [9]

Aus Abbildung 3 wird ersichtlich, dass für die Verschiebung der Drehmoment-Drehzahl-Kennlinie nicht nur die Frequenz f erhöht wird, sondern auch im gleichen Maß die Drehspannung U_{Dreh} . Durch die Verschiebung der Drehmoment-Drehzahl-Kennlinie nach rechts, kann der Asynchronmotor bei großen Lastmomenten höhere Drehzahlen erreichen. Die drei Strom-Phasenversetzungen φ_1 , φ_2 und φ_3 weisen standardmäßig eine gleichmäßige Versetzung von 120° untereinander auf. Durch das Verstellen einer Strom-Phasenversetzung kann beim Asynchronmotor ein Strom-Phasenfehler erzeugt werden. Auf Basis der Frequenz f und der Phasenversetzungen φ_1 , φ_2 und φ_3 erzeugt der Frequenzumrichter-Block eine entsprechende dreiphasige Drehspannung U_{Dreh} und gibt diese am Ausgang aus. Die Drehspannung U_{Dreh} wird zum Asynchronmotor übertragen.

Der Asynchronmotor wird im Simulink-Modell durch einen Asynchronmotor-Block aus der Simscape-Bibliothek simuliert. Dieser trägt die Modellbezeichnung 100 HP und weist eine Leistung von 75 kW auf. Als Eingangsparameter erhält der Asynchronmotor-Block die Drehspannung U_{Dreh} vom Frequenzumrichter und das Gesamt-Lastmoment bzw. Motor-Drehmoment M_{LM} , welches sich aus den einzelnen Lastmomenten im gesamten Simulations-Modell zusammensetzt. Der Asynchronmotor-Block wandelt die elektrische Energie in mechanische Energie und gibt am Ausgang die Motor-Drehzahl n_1 , die Stator-Ströme I_S und Rotor-Ströme I_R aus. Die Motor-Drehzahl n_1 wird zum Riementrieb-Block übertragen.

Der Riementrieb-Block erhält als Eingangsparameter das Motorwellen-Lastmoment M_{LM} , das Ventilator-Lastmoment M_{LV} , den Schlupf s , die Riemenscheiben-Radien r_1 und r_2 , die Reibungszahl μ , den Umschlingungswinkel von der Motor-Riemenscheibe β , die Riemenscheiben-

Exzentrizität e_R und die Motor-Drehzahl n_1 . Im Riementrieb-Block wird auf Basis der eingestellten Riemenscheiben-Radien r_1 und r_2 , des Schlupfs s und der Motor-Drehzahl n_1 die Ventilator-Drehzahl n_2 gebildet.

$$n_2 = (1 - s) \cdot \frac{n_1}{\left(\frac{r_2}{r_1}\right)} \quad (1)$$

Durch Verstellung der Riemenscheiben-Exzentrizität e_R kann eine sinusförmige Unwuchtbeschleunigung an der Motor-Riemenscheibe a_{UR} simuliert werden. Die sinusförmige Unwuchtbeschleunigung a_{UR} wird auf Grundlage der Riemenscheiben-Exzentrizität e_R und der Motor-Drehzahl n_1 generiert.

$$a_U = e_R \cdot (2 \cdot \pi \cdot n_1)^2 \quad (2)$$

Je größer die Riemenscheiben-Exzentrizität e_R ist, umso größer wird die Amplitude der Unwuchtbeschleunigung an der Motor-Riemenscheibe a_{UR} . Die Frequenz der Unwuchtbeschleunigung a_{UR} wird durch die Motor-Drehzahl n_1 beeinflusst. Außerdem werden auf Basis des Motor-Drehmoments M_{LM} , des Ventilator-Lastmoments M_{LV} , die Riemenscheiben-Radien r_1 und r_2 , des Umschlingungswinkels von der Motor-Riemenscheibe β und der Reibungszahl μ die Wellenbelastungen an der Motorwelle F_{WM} und der Ventilatorwelle F_{WV} gebildet.

$$F_W = \frac{M_L}{r} \cdot \frac{\sqrt{(e^{\mu\beta})^2 + 1 - 2 \cdot (e^{\mu\beta}) \cdot \cos(\beta)}}{(e^{\mu\beta}) - 1} \quad (3)$$

Am Ausgang des Riementrieb-Blocks wird die Ventilator-Drehzahl n_2 , die Unwuchtbeschleunigung a_{UR} an der Motor-Riemenscheibe und die jeweilige Wellenbelastung an der Motorwelle F_{WM} und der Ventilatorwelle F_{WV} ausgegeben.

Der Ventilator-Block erhält als Eingangsparameter den Ventilator-Liefergrad λ , den Ventilator-Wirkungsgrad η , den Ventilator-Außendurchmesser D , die Ventilator-Exzentrizität e_V und die Ventilator-Drehzahl n_2 . Der Liefergrad λ beschreibt das Verhältnis der tatsächlichen Fördermenge zur theoretisch möglichen Fördermenge. Durch die Verringerung des Ventilator-Liefergrads λ und des Ventilator-Wirkungsgrads η können Schmutzablagerungen im Ventilator-Vorfilter und am Ventilator simuliert werden. Auf Basis des Ventilator-Liefergrads λ , des Ventilator-Wirkungsgrads η , des Ventilator-Außendurchmessers D , der Lüfter-Drehzahl n_2 und der implementierten Ventilator-Kennlinie stellt sich ein entsprechender

Volumenstrom Q und eine entsprechende Totaldruckerhöhung Δp_{Total} zwischen der Saug- und Druckseite des Ventilators ein.

$$Q = \frac{\lambda \cdot \pi^2 \cdot D^3 \cdot n_2}{4} \quad (4)$$

Die Ventilator-Kennlinie beschreibt das Betriebsverhalten des Ventilators als Funktion des Förderstroms bzw. Volumenstroms Q . Deren Ermittlung erfolgt auf einem Prüfstand durch Eindrosseln verschiedener Volumenströme [10]. Dabei wird meist die Drehzahl des Ventilators konstant gehalten [10]. Der Betriebspunkt des Ventilators stellt sich am Schnittpunkt zwischen Anlagen- und Ventilator-Kennlinie ein. In Abbildung 4 ist eine solche Anlagen- und Ventilator-Kennlinie dargestellt.

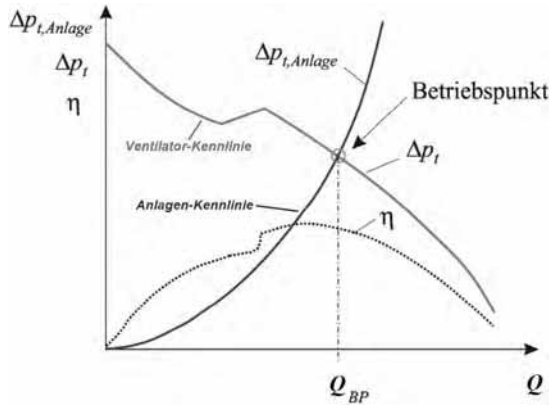


Abbildung 4: Aerodynamische-Kennlinien [10]

Neben dem Volumenstrom Q und der Totaldruckerhöhung Δp_{Total} beeinflusst der Betriebspunkt des Ventilators auch das Ventilator-Lastmoment M_{LV} .

$$M_{LV} = \frac{Q \cdot \Delta p_{Total}}{2 \cdot \pi \cdot n_2 \cdot \eta} \quad (5)$$

Des Weiteren kann mittels Verstellung der Ventilator-Exzentrizität e_V eine durch Schmutzablagerungen verursachte sinusförmige Unwuchtbeschleunigung am Ventilator-Lager a_{UV} simuliert werden. Die sinusförmige Unwuchtbeschleunigung a_{UV} wird auf Grundlage der Ventilator-Exzentrizität e_V und der Ventilator-Drehzahl n_2 generiert. Je größer die Ventilator-Exzentrizität e_V ist, umso größer wird die Amplitude der Unwuchtbeschleunigung am Ventilator-Lager a_{UV} . Die Frequenz der Unwuchtbeschleunigung a_{UV} wird durch die Ventilator-Drehzahl n_2 beeinflusst. Am Ausgang des Ventilator-Blocks wird das Ventilator-Lastmoment M_{LV} , der Volumenstrom Q , die Totaldruckerhöhung Δp_{Total} und die Unwuchtbeschleunigung a_{UV} ausgegeben.

Das Motor-Lastmoment M_{LM} am Asynchronmotor ergibt sich aus der Summe von dem Lagerreibungsmoment an der Motorwelle M_{RMW} , dem übersetzten Lagerreibungsmoment an der Ventilatorwelle $M_{RVW/i}$ und dem übersetzten Ventilator-Lastmoment $M_{LV/i}$. Die Lagerreibungsmomente werden jeweils auf Basis der Lager-Kenngrößen für die Motorwelle L_M und Ventilatorwelle L_V , des drehzahlabhängigen Lagerbeiwerts f_0 , des lastabhängigen Lagerbeiwerts f_i , der Wellenbelastung an der Motorwelle F_{WM} und der Ventilatorwelle F_{WV} und der Elementabstände an der Motorwelle EM und der Ventilatorwelle EV gebildet. Zu den Lager-Kenngrößen zählen der Lager-Mitteldurchmesser d_m , die Lager-Tragzahl C_0 und die Viskosität des Schmieröls ν . Durch die Erhöhung der Lagerbeiwerte f_0 und f_i kann Lagerverschleiß simuliert werden.

Im Rahmen von Simulationen werden im Simulink-Modell der Lüftungsanlage die Werte von den Unwuchtbeschleunigungen, von den Rotor- und Stator-Strömen im Asynchronmotor, vom Motor-Drehmoment, von der Motor- und Ventilator-Drehzahl, vom Volumenstrom und von der Gesamt-Stromaufnahme des Asynchronmotors in eine MAT-Datei exportiert, damit diese für nachgelagerte ML-Ansätze verwendet werden können.

3 Evaluation des Simulationsmodells

Die Evaluation des Simulationsmodells zur Lüftungsanlage erfolgt durch die Simulation von Fehlerzuständen und der Auswertung der dabei aufgezeichneten Messgrößenverläufe, welche sich aufgrund der Wirkzusammenhänge einhergehend mit dem Fehler verändern sollen.

Zunächst wird im Simulationsmodell ein Schlupf am Riementrieb simuliert. Dabei wird erwartet, dass aufgrund des Schlupfs am Riementrieb die Ventilator-Drehzahl und der Volumenstrom absinkt. In Abbildung 5 ist der Verlauf von der Ventilator-Drehzahl und vom Volumenstrom über die Simulations-Zeit zu sehen. Im Zeitintervall zwischen 0 s und 2,8 s erfolgt der Anfahrvorgang des Ventilators. Dabei steigt die Ventilator-Drehzahl von 0 min^{-1} auf 640 min^{-1} und der Volumenstrom von 0 m^3/s auf 7,9 m^3/s exponentiell an. Bei einem gleichbleibenden Schlupf von 0% bleibt im Zeitabschnitt zwischen 2,8 s und 5 s die Ventilator-Drehzahl bei 640 min^{-1} und der Volumenstrom bei 7,9 m^3/s annähernd konstant. Nach 5 s wird der Schlupf sprunghaft auf 5% erhöht. Dabei sinkt die Ventilator-Drehzahl auf ca. 610 min^{-1} und der Volumenstrom auf etwa 7,5 m^3/s sprunghaft ab. Die zuvor beschriebenen Messgrößenverläufe

haben sich nach Eintreten des Fehlerzustands wie erwartet verhalten.

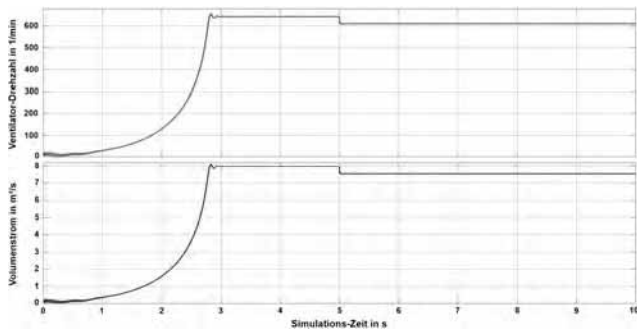


Abbildung 5: Verlauf von Ventilator-Drehzahl und Volumenstrom über die Simulationszeit

Nachfolgend werden im Simulationsmodell Schmutzablagerungen im Ventilator-Vorfilter und am Ventilator simuliert. Dabei wird erwartet, dass der Betriebspunkt von der Lüftungsanlage wie in Abbildung 6 auf der Ventilator-Kennlinie nach oben wandert und somit der Volumenstrom sinkt, die Totaldruckerhöhung steigt, der Ventilator-Wirkungsgrad sich geringfügig verändert und eine größere Unwuchtbeschleunigung am Ventilator-Lager entsteht.

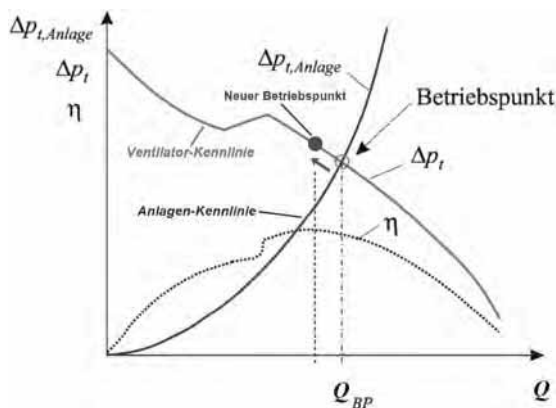


Abbildung 6: Verschiebung des Betriebspunkts vom Ventilator [10]

In Abbildung 7 ist der Verlauf vom Volumenstrom, von der Totaldruckerhöhung, vom Ventilator-Wirkungsgrad und von der Unwuchtbeschleunigung am Ventilator-Lager dargestellt. Im Zeitintervall zwischen 0 s und 2,8 s erfolgt der Anfahrvorgang des Ventilators. Dabei steigt der Volumenstrom von 0 m³/s auf 7,9 m³/s exponentiell an, während die Totaldruckerhöhung von 2580 Pa auf 2090 Pa exponentiell absinkt und der Ventilator-Wirkungsgrad von 52 % auf 79 % exponentiell ansteigt. Die Amplitude der Unwuchtbeschleunigung am Ventilator-Lager liegt in dieser Zeit nahezu bei 0 m/s². Bei einem sauberen Ventilator-Vorfilter und sauberen Ventilator

bleibt im Zeitabschnitt zwischen 2,8 s und 5 s der Volumenstrom bei 7,9 m³/s, die Totaldruckerhöhung bei 2090 Pa und der Ventilator-Wirkungsgrad bei 79 % annähernd konstant, während die Unwuchtbeschleunigung am Ventilator-Lager annähernd sinusförmig mit einer Amplitude von 0,4 m/s² und einer Frequenz von 11 Hz oszilliert. Nach 5 s werden die Schmutzablagerungen im Ventilator-Vorfilter und am Ventilator linear vergrößert, indem jeweils der Liefergrad und der Wirkungsgrad des Ventilators abgesenkt wird. Dabei sinkt der Volumenstrom auf 7,1 m³/s und der Ventilator-Wirkungsgrad auf 76 % ab während die Totaldruckerhöhung auf 2190 Pa steigt. Die Amplitude der Unwuchtbeschleunigung am Ventilator-Lager ist aufgrund der ungleichmäßigen Verteilung der Schmutzablagerungen am Ventilator und der dadurch entstehenden erhöhten Exzentrizität auf 3,8 m/s² angestiegen. Die Frequenz der Unwuchtbeschleunigung hat sich nicht verändert, da sich die Ventilator-Drehzahl kaum verändert hat. Die zuvor beschriebenen Messgrößenverläufe haben sich nach Eintreten des Fehlerzustands wie erwartet verhalten und erscheinen daher plausibel.

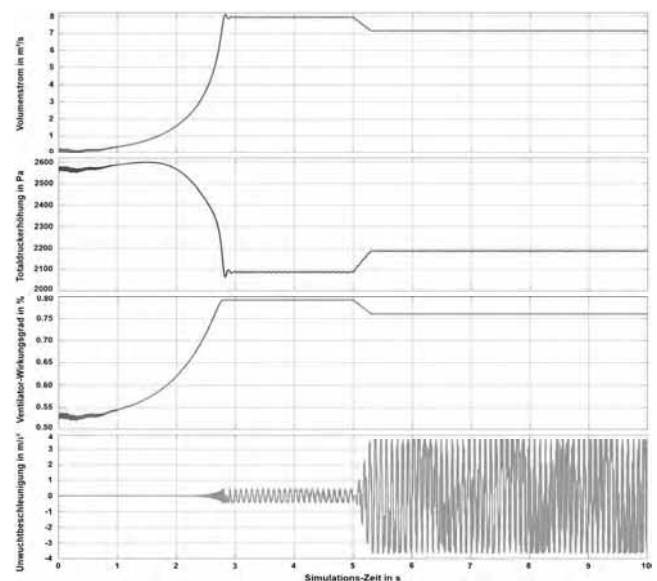


Abbildung 7: Verlauf von Volumenstrom, Totaldruckerhöhung, Ventilator-Wirkungsgrad und Unwuchtbeschleunigung über die Simulationszeit

4 Einsatzmöglichkeiten für ML-Ansätze

Das Ziel dieser Arbeit liegt nicht darin, den digitalen Zwilling einer real existierenden Lüftungsanlage exakt abzubilden. Der Fokus liegt darauf, insbesondere Ingenieuren und Studierenden der Ingenieurwissenschaften den

Einstieg in die Anwendung von ML-Ansätzen für die Zustandsüberwachung von Maschinen und Anlagen zu erleichtern.

Mithilfe des Simulationsmodells lassen sich Fehlerbilder in verschiedensten Ausprägungen darstellen und die zugehörigen Messdaten erzeugen. Dies erleichtert den Zugang zu den Zusammenhängen zwischen den unterschiedlichen Ausprägungen verschiedenster Fehlerbilder und den daraus resultierenden Veränderungen in den Messdaten. Dieses Verständnis erleichtert die Auswahl geeigneter Ansätze für das Training von ML-Modelle für das Erkennen bestimmter Fehlerbilder bzw. der dadurch verursachten Anomalien in den Messdaten erheblich. Ein weiterer Nutzen entsteht dadurch, dass Anwender das Simulationsmodell für die Evaluierung der auf Basis der künstlich erzeugten Messdaten entwickelten ML-Modell nutzen können.

Diese Möglichkeit ist insbesondere interessant, um die Robustheit entwickelter Modelle hinsichtlich variierender Betriebsparameter zu untersuchen. Auch Fragestellungen zur Übertragbarkeit entwickelter ML-Modelle auf bauähnliche Anlagen lassen sich durch die Veränderung einzelner Parameter des Simulationsmodells am praktischen Beispiel testen. Des Weiteren erleichtert die experimentelle Arbeit mit verschiedenen Trainingsdatensätzen, die Einschätzung der Eignung einer im konkreten Anwendungsfall vorhandenen Datenbasis für geplante ML-Ansätze.

Zusammenfassend lässt sich das entwickelte Simulationsmodell als hilfreiches Werkzeug für den Einstieg in eine ingenieurwissenschaftliche Herangehensweise an das Thema ML zur Zustandsüberwachung von Maschinen und Anlagen einsetzen.

5 Ausblick

Aktuell beschäftigen sich sowohl Unternehmen als auch Forschungseinrichtungen mit der Generierung von Trainingsdaten für ML-Ansätze, da sie ein großes Potenzial in diesem Ansatz sehen, um den Problemen beim Machine Learning entgegenzuwirken [11]. Das hier vorgestellte Simulationsmodell soll insbesondere für Vorhaben in Forschung und Lehre in diesem Themengebiet einen Beitrag leisten. Darüber hinaus soll im Rahmen der Zustandsüberwachung von Maschinen und Anlagen die Eignung von Simulationsmodellen für die Generierung von Trainingsdaten und die Übertragbarkeit der damit trainierten ML-Modelle auf ein reales System untersucht werden.

Literaturen

- [1] Döbel, I. et al: *Maschinelles Lernen*. Eine Analyse zu Kompetenzen, Forschung und Anwendung, Fraunhofer-Gesellschaft, 2018.
- [2] *IDG Studie Machine Learning / Deep Learning 2019*, IDG Business Media GmbH, 2019.
- [3] BMBF: *Erzeugung von synthetischen Daten für Künstliche Intelligenz*, bmbf.de, 04.06.2020.
- [4] A. von Birgelen, A. Fast: *Was Sie beim Einsatz von Machine Learning beachten müssen*, industr.com, 11.05.2020.
- [5] N. Funke: *Predictive Maintenance auf dem Prüfstand*, agile-im.de, 22.09.2020.
- [6] acatech POSITION: *Smart Maintenance für Smart Factories*, acatech, 2015.
- [7] M. Luft: *Elektromaschinen-Diagnose mit Kennwerten und FFT-Analyse*, dbprüftechnik.
- [8] S. Luber, N. Litzel: *Was ist Simulink?*, www.bigdata-insider.de, 17.01.2019.
- [9] Danfoss: *Wissenswertes über Frequenzumrichter*, 2015.
- [10] T. Carolus: *Ventilatoren*, 4. Auflage, Berlin Heidelberg: Springer-Verlag, 2020.
- [11] M. Göbel: *Große Hoffnung für späte Starter bei künstlicher Intelligenz*, basecamp.digital, 09.08.2018.

Systems Engineering als Basis für konstruktionsbegleitende Zusammenarbeit

Eva Russwurm^{1*}, Florian Faltus¹, Joerg Franke¹

¹Lehrstuhl für Fertigungsautomatisierung und Produktionssystematik, FAU Erlangen-Nürnberg, Egerlandstraße 7-9, 91058 Erlangen, Deutschland; *eva.russwurm@faps.fau.de

Abstract. Der Ansatz des Systems Engineering (SE) unterliegt einem stetigen Wandel und wird bereits in vielen Unternehmen im Rahmen der Produktentwicklung eingesetzt. Dieser Ansatz, der in Zusammenarbeit, Effizienz und Produktqualität, sowie Kosten viele Vorteile bietet, kann den Herausforderungen der Industrie 4.0 begegnen. Zur Umsetzung des SE werden verschiedene Werkzeuge benötigt. Dazu gehören neben einem PLM-System zur entwicklungsbegleitenden Zusammenarbeit auch verschiedene Simulationsumgebungen, die zu einer Co-Simulation zusammengeführt werden. Eine Möglichkeit, hierfür Kommunikation zu ermöglichen, bietet hier die Verwendung des Kommunikationsstandards OPC UA. Die Anwendung des SE-Ansatzes soll anhand von zwei Anwendungsbeispielen dargestellt werden, nämlich anhand einer ganzheitlichen Simulation einer Produktionsanlage mit Energiemanagement und in einer Lehrveranstaltung.

Einleitung

Die Automatisierung von Produktionssystemen ist durch eine hohe technische Komplexität und starke Interdisziplinarität gekennzeichnet. Die Planung einer Industrieanlage erfordert die Koordination und Integration verschiedener Fachdisziplinen wie Mechanik, Prozess- und Elektrotechnik oder Software in Bezug auf das Vorgehen und die Arbeitsergebnisse.

In diesem Zusammenhang stellt die Automatisierung als Bindeglied das korrekte Zusammenwirken der verschiedenen Disziplinen sicher. Durch den Ansatz Industrie 4.0 und der damit verbundenen Zunahme der IT-Durchdringung in der produzierenden Industrie steigt die Relevanz der Digitalisierung der Fertigung deutlich.

Dies ist schon während der Produktentwicklung ersichtlich, denn bereits hier arbeiten die Ingenieure unterschiedlicher Disziplinen zusammen im Rahmen des Systems Engineering. Hierzu braucht es eine Datenplattform, über welche eine Konsistenz der Daten sichergestellt und Zugriffe ermöglicht werden, sowie eine Möglichkeit des Austauschs über alle Disziplinen hinweg.

In diesem Artikel soll zunächst ein Überblick über die Methodik des Durchgängigen Engineerings gegeben

werden und die Entwicklungsmethodik nach dem V-Modell eingeordnet werden. Im Anschluss daran soll aufgezeigt werden, wie die einzelnen Disziplinen synchron an einer Simulation teilnehmen und damit eine virtuelle Inbetriebnahme gelingen kann. Schließlich soll die Anwendung des durchgängigen Engineerings anhand von zwei Anwendungsfällen dargestellt werden; zum einen der methodische Ansatz in der Lehre mit Studierenden als Basis für zukünftige Zusammenarbeit im Unternehmen und zum anderen der Aufbau einer Co-Simulation für eine automatisierte Produktionsanlage in einem Industrieunternehmen.

1 Durchgängiges Engineering

Im modernen Maschinenbau gibt es kaum noch rein mechanische Produkte. Der Anteil an Elektronik und Software bzw. Maschinensteuerung im Produktentwicklungsprozess (PEP) steigt nicht nur kontinuierlich an, siehe Abbildung 1, sondern wird auch immer komplexer. Während in den 1980er Jahren eine Industrieanlage vorwiegend aus Mechanik bestand und die Mechatronisierung eher unterstützendes Beiwerk bildete, so ist es heutzutage die Software, die maßgeblich die Funktionalitäten einer Anlage bestimmt. Die Gründe hierfür entstehen aus den Anforderungen des Kunden wie Prozessflexibilisierung und Vernetzung aller Anlagen im Betrieb, um einfache Bedienung und Beobachtung der Produktion zu ermöglichen. [1]

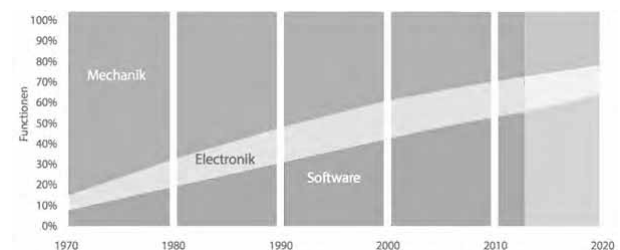


Abbildung 1: Anteil der Einzeldisziplinen im PEP [1]

Daher erfordert bereits eine erfolgreiche Produktentwicklung ein erfolgreiches Zusammenwirken der verschiedenen Kompetenzfelder Maschinenbau, Elektrotechnik und Informatik/Software Engineering. Diese Komplexität kann nicht mehr durch eine kleine Anzahl an Personen abgedeckt werden, sodass Entwicklungsteams interdisziplinärer und größer aufgestellt sein müssen, als dies in der Vergangenheit der Fall war. Dies führt unweigerlich dazu, dass die Gruppen auch an verschiedenen Orten in unterschiedlichen Zeitzeonen zusammenarbeiten müssen.

1.1 Definition mechatronischer Systeme

Der Begriff „Mechatronik“ ist ein Kunstwort aus Mechanik und Elektronik und bezeichnet „das synergetische Zusammenwirken der Fachdisziplinen Maschinenbau, Elektrotechnik und Informationstechnik beim Entwurf und der Herstellung industrieller Erzeugnisse, sowie bei der Prozessgestaltung“ [2]. Das interdisziplinäre Zusammenspiel einer hohen Anzahl verkoppelter Elemente, Schnittstellen und Wechselwirkungen der mechatronischen Module untereinander macht die Anwendung eines ganzheitlichen, disziplinübergreifenden Ansatzes zur Systemdefinition unabdingbar. Hierzu zählen auch Kommunikation und Kooperation zwischen den einzelnen Fachbereichen [3].

Aus diesem Grund kommen für mechatronische Systeme immer häufiger neue Entwicklungsmethoden, die auf dem Grundgedanken des Systems Engineering (SE) basieren, zum Einsatz.

1.2 Strategie des Durchgängigen Engineerings

Die drei Leitgedanken des Simultaneous Engineering sind Parallelisieren, Standardisieren und Integrieren. Dabei bedeutet Parallelisieren zeitliche Optimierung von Teilprozessen, die unabhängig voneinander sind und bearbeitet werden können. Im Bereich der Standardisierung ist es das vorrangige Ziel, Mehr- und Wiederholarbeiten zu vermeiden. Dies gelingt vor allem durch eine vorgegebene einheitliche Gestaltung von Modulen, Komponenten, Phasen und Schnittstellen zwischen Projekten und Abteilungen. Das Ziel der Integration ist es, die Schnittstellen zu Nahtstellen zu machen. Sie dienen der Zusammenführung aller Produktinformationen.

1.3 Multidisziplinärer Ansatz und Parallelisierung von Entwicklungstätigkeiten

Diesem multidisziplinären Aufbau des Modells liegt ein iteratives Vorgehen zugrunde. Integrationsfortschritt und Synchronisationspunkte werden dabei fortwährend geprüft und garantieren eine erfolgreiche Produktentwicklung, bedingt durch frühzeitige Fehlervermeidung.

Weiterhin kann durch die Parallelisierung anstelle einer sequentiellen Produktentwicklung wertvolle Zeit eingespart werden, was sich positiv auf das geforderte Zeitziel (Time-to-market) auswirkt. Damit dieses Ziel eingehalten werden kann, müssen bestimmte Methoden, wie beispielsweise Wiederverwendung innerhalb des SEs angewandt werden [4].

Folglich ist die Verwendung von SE zur Unterstützung des aufkommenden Wandels zu individuellen Produkten als Lösungsansatz zu betrachten.

2 Produktentwicklung entlang des V-Modells

2.1 Anforderung- und Lösungsspezifikation

Ein weiteres Konzept, welches vor allem im Bereich des Requirement Engineering, d.i. die Entwicklung der Anforderungen aus abstrakten Vorgaben des Kunden, genutzt wird, ist der RFLP-Ansatz [5]. In deutschsprachiger Literatur auch AFLP-Ansatz genannt, beschreibt es die Unterteilung in Anforderungen/ Requirements, Funktionen, Logik und physisches Modell. Durch diese vier Bereiche kann das Ziel des Konzepts, nämlich der Entwurf einer einheitlichen Struktur, die auf alle Disziplinen angewendet werden kann, verfolgt werden. Diese interdisziplinäre Zusammenführung resultiert in einer zunächst disziplinunabhängigen Beschreibung des Systems und nimmt eine außerordentliche Rolle besonders in den frühen Entwicklungsphasen von multidisziplinären Produkten ein.

Das physikalische Modell beschreibt die Ausarbeitung der Systemarchitektur durch die Ergänzung physikalischer Eigenschaften unter dem Einsatz von disziplinspezifischen Methoden und IT-Anwendungen [6].

2.2 Mechanik-, Elektrik- und Softwareentwicklung

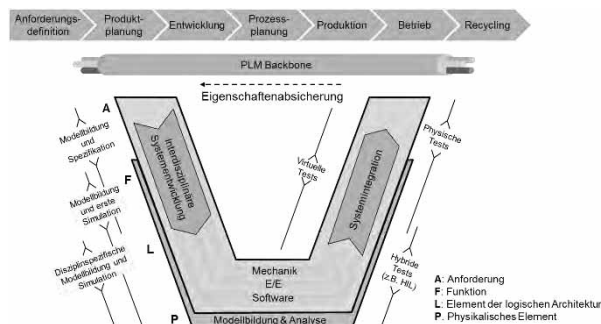


Abbildung 2: MVPE Vorgehensmodell auf Basis VDI 2206 [3]

Im Prozess des Systems Engineering folgt die mechanische Konstruktion auf die Lösungsspezifikation. Simultan mit ihr laufen die Elektrokonstruktion sowie die Softwareentwicklung, wie aus Abbildung 2 ersichtlich. Nach Abschluss der Konstruktionsphase, derer sie angehört, folgt die Realisierung sowie die virtuelle und die tatsächliche Inbetriebnahme [3].

Um nun einen durchgängigen Informationsfluss und ein disziplinübergreifend stimmiges Produktmodell zu ermöglichen, besitzt die jeweilige Disziplin naturgemäß Schnittstellen mit anderen Disziplinen (Mechanik, Elektrik und Software). Dazu müssen eigens neue Ansätze der Produktentwicklung definiert werden [7]. Die Schnittstellen werden vorwiegend durch das PLM-System abgebildet.

2.3 Weiterentwicklung zum modellbasierten Systems Engineering

Ein möglicher Ansatz ist das Modellbasierte Systems Engineering (MBSE), welches den Übergang des dokumenten- zum modellbasierten Systems Engineerings beschreibt und die MVPE mit dem Systems Engineering vereint. Es handelt sich hierbei um eine „formalisierte Anwendung der Modellierung, um die Aufnahme von Systemanforderungen, -design, -analyse, Verifikation und Validierung ab der Konzeptphase über die Entwicklungsphase bis zu späteren Lebenszyklusphasen zu unterstützen“ [8]. In diesem Fall kommt der Entwurfsphase eine besondere Bedeutung zu, da hier das Systemmodell erstellt wird, welches sämtliche Produktlebenszyklusanforderungen des Produktes beinhaltet.

Bereits während des Produktentwicklungsprozesses entstehen vollständig digitale Produktmodelle (Anlagenmodelle), die eine Verknüpfung der physischen Produkte

mit den zugehörigen virtuellen Modellen aus der Produktentwicklung ermöglichen. Damit gelingt die in der Industrie 4.0 geforderte Vernetzung realer Produktionseinrichtungen mit den in der Produktplanung entstandenen digitalen Abbildungen [10], die dazu beitragen, dass virtuelle und reale Welt miteinander verschmelzen.

Im Rahmen des MBSE entsteht für jedes Produkt ein holistisches und konsistentes Datenmodell über den gesamten Produktlebenszyklus, das enorme Produktivitäts-, Effizienz- und Qualitätssteigerungen ermöglicht, da reale und digitale Prozesse überlappend stattfinden. Beispielsweise wird neben dem Produkt auch die Fertigung digital geplant und durch frühzeitige Simulationen abgebildet und validiert, was zu einer um bis zu 50% verringerten Markteinführungszeit neuer Produkte führt [11]. Das durchgängige Datenmodell ermöglicht durch die Konsistenz Flexibilität im Produktionsprozess, die eine Individualisierung von Produkten ermöglicht.

2.4 Eingliederung des MBSE in den Stage-Gate Prozess und das Quality Gate Modell

Das VDMA Quality Gate-Modell entspricht einem Stage-Gate Prozess nach dem Vorbild Coopers [9]. Er unterteilt Innovations- bzw. Produktentwicklungsprozesse in verschiedene Phasen (englisch: stages), zu welcher jeweils ähnliche Aktivitäten zusammengefasst werden. Die einzelnen Phasen enden mit einer Qualitätskontrolle (Gate), das nur überschritten werden kann, wenn die definierten Anforderungen erfüllt sind. Ausschlaggebend ist hierzu die Einschätzung des Managements. Ein Vorteil dieser Methode ist in der Strukturierung langer Prozesse, die dadurch kontrollier- und steuerbar werden, zu finden. Der regelmäßige Abgleich des Projektteams mit dem Management dient außerdem der Information aller Stakeholder und bezieht schon früh im Prozess interdisziplinäre Unternehmensbereiche (Marketing, Vertrieb etc.) ein. Insgesamt führen Stage-Gate Prozesse zu höherer Qualität und mehr Innovationen [9]; [12].

Das Vorgehen weist viele Parallelen zum V-Modell der VDI Norm 2206 auf und beide Modelle umfassen nahezu die gleichen Schritte. Die Beschreibung der Phaseninhalte, Deliverables und Gate-Kriterien machen aus dem allgemeinen V-Modell einen steuerbaren Prozess. Hier wird jedoch noch nicht modellbasiert gearbeitet, denn die größte Herausforderung ist die Realisierung der Durchgängigkeit im PEP mittels konsistenter Systemmodelle.

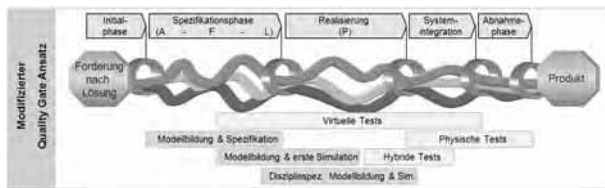


Abbildung 3: Modellbasierter Quality Gate Ansatz

Abbildung 3 stellt den veränderten Quality-Gate Ansatz dar. Die türkisen, gelben und roten Bänder repräsentieren hier ebenso wie im VDMA Quality-Gate Modell die klassischen Disziplinen Mechanik, Elektronik und Software, welche weiterhin als parallel verlaufende Stränge bearbeitet werden. [13]; [14]

Im Rahmen der Weiterentwicklung werden Anforderungs- und Lösungsspezifikationsphase zu einer einzigen Spezifikationsphase zusammengefasst, so dass ein ganzheitliches Systemmodell aus Systemanforderungen, Funktionen und Logik (A, F, L) erstellt und freigegeben werden kann. Auch die Erstellung von Testspezifikationen bildet einen Teil dieser Phase. Die Durchführung quantitativer Simulationen unterstützt zudem die objektive Auswahl von Lösungsalternativen.

Die anschließende Realisierungsphase beinhaltet die disziplinspezifische Modellbildung und Simulation, die spätestens in der Integrationsphase zusammengeführt und durch physische Tests ergänzt wird. Abschließende Tests werden in der Abnahmephase durchgeführt.

Wird während des PEP mit sich weiterentwickelnden Modellen gearbeitet, ist die Simulation als unterstützendes Modul von großer Bedeutung. Weiterhin bilden die ersten Simulationsstudien, die, wie in Abbildung 3 dargestellt, bereits in der Spezifikationsphase starten, die Grundlage für den digitalen Zwilling des Produktes bzw. der Produktionsanlage [15].

3 CO-Simulation zur Synchronisierung der Disziplinen

„Simulation ist das Nachbilden eines Systems mit seinen dynamischen Prozessen in einem experimentierfähigen Modell, um zu Erkenntnissen zu gelangen, die auf die Wirklichkeit übertragbar sind.“ [16, S. 3]

3.1 Simulation von Produktionssystemen

Heutzutage ist es notwendig, ein automatisiertes Produktionssystem nicht nur in der Mechanik (z. B. mittels Fi-

nite-Elemente-Simulation) hinsichtlich Festigkeit auszuwerten, sondern das gesamte mechatronische Modell zu betrachten und Wechselwirkungen zwischen Modulen und Modellen zu berücksichtigen, denn es handelt sich dabei um ein komplexes Gebilde. Die genannte Komplexität steigt mit der Zahl der Elemente ebenso wie durch die Verknüpfungen zwischen den Elementen hinsichtlich Toleranz oder Dynamik. Um die Komplexität beherrschbar zu gestalten, gliedert man das System in mehrere – teils hierarchische – Ebenen. In jeder Ebene können relevante Fragestellungen durch die Simulation beantwortet werden.

Dabei gilt, dass zunächst die Fragestellungen und Lösungen in den einzelnen Ebenen betrachtet, ausgewertet und ggf. verbessert werden. Dies geschieht unabhängig von Simulationen anderer Ebenen. Eingeordnet in den Einsatz des MBSE bedeutet dies, dass sowohl einzelne Baugruppen der Anlage getrennt, aber auch, dass die einzelnen Disziplinen unabhängig voneinander betrachtet werden können. Dass der zweitgenannte Ansatz keine sinnvollen Lösungen ergeben wird, ist spätestens bei Betrachtung mechatronischer Modelle erwiesen, denn mechanische, elektrische und informationstechnische (Software) Aufgaben müssen integriert gelöst werden.

Insgesamt hat sich die Simulation als eine gewinnbringende Methode erwiesen, Diskussionspunkte aufzuzeigen, aber auch komplexe Entscheidungsprozesse zu unterstützen. [17] In der Simulationstechnik liegen also große Chancen zur Verbesserung bei Planung und Betrieb von Produktionssystemen [18].

3.2 Multiphysikalische Simulationsprogramme

Von diversen Herstellern gibt es bereits Simulationsprogramme, die für die Lösung mechatronischer Anforderungen prädestiniert sind. Beispiele hierfür bilden der Mechatronics Concept Designer (MCD) von Siemens PLM, iPhysics von machineering oder virtuos von isg. Diese Systeme bieten nicht nur eine Simulationsumgebung, sondern auch Schnittstellen zur virtuellen Inbetriebnahme [19].

Es handelt sich hierbei um Werkzeuge für mechanische und elektromechanische Simulation (Computer-Aided-Engineering), Fertigung, Werkzeug- und Vorrichtungsbau, Qualitätsprüfung und mechatronische Konzeptentwicklung [20]. Dabei können in der Mechanikkonstruktion modellierte Geometrien zu einem Simulationsmodell erweitert werden, das neben der mehrkörperphysikbasierten Simulation auch Aspekte der Automatisie-

rungstechnik beinhaltet und so das physikalische Verhalten unterschiedlicher Lösungskonzepte visualisiert. Ausgehend vom Ansatz des MBSE und einem domänenübergreifenden Lösungskonzept fördern die Simulationstools eine frühzeitige interdisziplinäre Zusammenarbeit von Mechanik, Elektronik und Softwareentwicklung. Dies spiegelt sich besonders durch Kostenersparnis und beschleunigte Produktentwicklungszeiten wider.

3.3 Kopplung von Simulationsprogrammen

Der genannte steigende Komplexitätsgrad von Produktionssystemen macht es erforderlich, weitere Simulationsumgebungen zu verknüpfen. Hierzu hat in den vergangenen Jahren die verteilte Simulation an Bedeutung gewonnen. Dabei handelt es sich zunächst um ein Simulationsmodell, das im Sinne von unterschiedlichen Ebenen auf unterschiedliche Modelle aufgeteilt wird. Zwischen den einzelnen Teilmodellen existieren Datenflüsse über Datenbanken, welche die Konsistenz dieser sicherstellen. Digitale Produktdaten in Form von Arbeitsplänen, Berechnungen und CAD-Modellen sind in verschiedenen Datenbanksystemen verfügbar und von allen am Entwicklungsprozess beteiligten Personen nutz- und veränderbar. Die entstehenden Produkt- oder Produktdatenmodelle, welche sowohl als fachübergreifende Informationsträger, als auch als Bindeglied zwischen den einzelnen Produktentstehungsbereichen, wie beispielsweise Planung und Konstruktion fungieren, bilden die Basis der verteilten Simulation.

Die verteilte Simulation wird auch als Co-Simulation bezeichnet [21] und kann, wie in Tabelle 1 aufgezeigt, aus einer partitionierten Simulation oder einer Werkzeugkopplung bestehen.

		Anzahl der Simulationstasks	
		= 1 geschlossene Simulation	> 1 verteilte Simulation
Anzahl der Modellierungs- werkzeuge	= 1 geschlossene Modellierung	monolithische Simulation	partitionierte Simulation Co-Simulation
	> 1 verteilte Modellierung	Modellkopplung	Werkzeugkopplung Co-Simulation

Tabelle 1: Matrix der Simulationsarchitekturen [21]

Die bereits genannten, im Kontext der Digitalen Fabrik entstandenen, disziplinspezifischen Simulationslösungen sind sehr leistungsfähig im Hinblick auf die Abbildungsgenauigkeit als auch auf die Berechnungsperformance. Allerdings sind diese für die virtuelle Abbildung ausgewählter Vorgänge ausgelegt.

Für eine umfassende Modellbildung des gesamten

Prozess-, Maschinen- und Anlagenverhaltens mit sämtlichen auftretenden Wechselwirkungen müssen hochauflösende Teilmodelle verschiedenster Simulationsdomänen in das digitale Abbild der Produktionsanlage integriert und in einer Gesamtsimulation gekoppelt werden. [22]

Im Kontext der multidisziplinären Modellierung von NC-Bearbeitungsmaschinen wird ein zentrales Problem in der Vereinigung von Modellen verschiedener Simulationsdisziplinen zu einem multidisziplinären Gesamtmodell genannt [23]. Denn grundsätzlich müssen die relevanten Daten der einzelnen Modelle über eine neutrale einheitliche Datenschnittstelle mittels eines einheitlichen Datenformats bereitgestellt werden. Hersteller von Simulationssoftware bestätigen mittlerweile diese These durch die Integration von neutralen Datenschnittstellen [24].

Weiterhin braucht es ein Simulationswerkzeug als Basis, das die Daten aller einzelnen Simulationswerkzeuge weitergibt. Hierzu gibt es verschiedene Ansätze und Tests in diversen Kontexten, jedoch kein System, das sich etabliert hat. In den hier aufgezeigten Anwendungsfällen soll ein Ansatz für eine Co-Simulation zur ganzheitlichen Simulation einer Produktionsanlage, die mit Gleichstrom betrieben wird, aufgezeigt werden.

4 IT-Infrastruktur

Zur Umsetzung des MBSE und damit verbunden der Simulation wird eine funktionsfähige und weitreichende Infrastruktur benötigt. Eine gängige Plattform zur sinnvollen Umsetzung des SE bilden PLM-Systeme, um das modellbasierte Arbeiten zu unterstützen und abzubilden. Zur Abbildung von Co-Simulationen werden weitere, möglichst echtzeitfähige Formate des Datenaustauschs benötigt.

4.1 PLM-System als Basis für das MBSE

Product Lifecycle Management (PLM) beschreibt einen „ganzheitlichen Ansatz zur unternehmensweiten Verwaltung und Steuerung aller Produktdaten und Prozesse des gesamten Lebenszyklus entlang der erweiterten Logistikkette“ [25].

Im Rahmen der Veränderung der Entwicklungsmethodik haben sich die Aufgaben des Ingenieurs von einer kreativen Entwicklungstätigkeit zu organisatorischen und kommunikativen Aufgaben (Planung, Beschaffung) verändert. Entsprechende IT-Werkzeuge zur effizienteren Bewältigung der Aufgaben sind notwendig [26] um die Digitalisierung zu beherrschen.

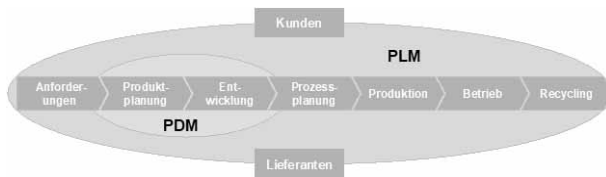


Abbildung 4: Abgrenzung zwischen PLM und PDM nach [27]

Diese Herausforderungen können nur durch den Einsatz von Product-Data-Management- (PDM) und PLM-Software bewältigt werden [27]. Erstere fokussieren die Phase der Produktentwicklung, während der Ansatz des PLM das Konzept erweitert, indem der gesamte Lebenszyklus von der ersten Idee bis zum Recycling betrachtet wird [28]. Demnach stellt PLM ein Konzept und nicht ein IT-System dar. Die PLM-Software ist eine Komponente des Konzeptes, dient der Integration von IT-Werkzeugen zu einer Entwicklungsumgebung und unterstützt das Zusammenspiel der Werkzeuge modell-, system-, prozess- und verfahrenstechnisch [3]. Ganzheitliche PLM-Konzepte ergeben sich aus Strategie, Prozessen und der IT-Lösung [32].

4.2 OPC UA als Kommunikationsstandard

Im Bereich der Automatisierung steht das Ziel effektiver, kostengünstiger und zeitsparender zu entwickeln und möglichst im Sinne von „First-time-right“ Anlagen direkt beim Kunden produktionsfähig in Betrieb zu nehmen, an erster Stelle. Ebenso wie der Bedarf nach Standardisierung und Modularisierung in der Software-Landschaft laut wird, entsteht eine erhöhte Nachfrage nach einer auf Kommunikationsebene standardisierten Schnittstelle zwischen mehreren Komponenten.

In diesem Zuge ist OPC entstanden. OPC steht für „Open Platform Communications“ und ist ein Kommunikationsprotokoll, das vor allem im Rahmen von Industrie 4.0 verwendet wird und das den standardisierten Zugriff auf Geräte, Maschinen und andere Systeme im industriellen Umfeld ermöglicht. Es bildet die Schnittstelle zwischen Steuerungssystemen und der Steuerungsebene, wodurch ein gleichmäßiger Datenaustausch unabhängig vom Hersteller ermöglicht wird. Die aktuellste Spezifikation stellt „Unified Architecture“ kurz UA dar. OPC UA besteht aus einem Server und dem Client, wobei der OPC UA-Server die Basis bildet. Das logische Gegenstück zum OPC UA-Server ist der Client. Durch das Verbinden mit dem Server können jene Daten ausgelesen werden, welche der Server bereitstellt. [29]; [30]

Mit Hilfe des beschriebenen OPC UA-Standards

wurde die Co-Simulation für eine automatisierte Produktionsanlage realisiert, welche mit einem eigenen Gleichstromkreis auf Industrieniveau mit einer Spannung von 650 V DC versorgt wird. Hier spielt neben der multiphysikalischen Simulation des Prozessablaufs die Simulation des Energieversorgungsnetzes eine bedeutende Rolle.

5 Virtuelle Inbetriebnahme als Systemintegration

Die Virtuelle Inbetriebnahme stellt ein Werkzeug der Digitalen Fabrik dar. Sie wird der Phase „Planung der Fertigungsanlagen“ zugeteilt und findet somit zeitlich vor der Phase „Montage und Inbetriebnahme der Fertigungsanlagen“ statt [31]. Sie bezeichnet die Steuerungsinbetriebnahme an einem virtuellen Maschinenmodell, welches die mechanischen, elektrischen, pneumatischen und hydraulischen Funktionalitäten einer automatisierten, mechatronischen Anlage abbildet [33].

Bei den drei Arten der virtuellen Inbetriebnahme (Model-in-the-loop (Mil), Software-in-the-loop (Sil) und Hardware-in-the-loop (Hil)) wird immer am Modell getestet. Es gilt also, das Modell so realitätsnah wie möglich zu modellieren. Die Begriffe „Model“, „Software“ und „Hardware“ beziehen sich in diesem Zusammenhang darauf, in welcher Form die Steuerung vorliegt, die verwendet wird, die Steuerung innerhalb des Modells, Steuerung durch eine simulierte Speicherprogrammierbare Steuerung (SPS) (über das Programm PLCSim Advanced) oder die Steuerung als Hardware-Komponente.

In jeder frühen Phase kann die Simulation nach Model-in-the-loop verwendet werden. Hier wird die Ablaufsteuerung direkt im physikalischen Modell eingefügt und innerhalb des Modells getestet. Dabei handelt es sich eher um eine prozessbegleitende Simulation, da sukzessive einzelne Schritte durch die Steuerung übernommen werden. An dieser Stelle wird aber bereits das physikalische Anlagenmodell verwendet, das auch bei den beiden anderen Formen der virtuellen Inbetriebnahme genutzt wird. In einem weiteren Schritt kann das Steuerungsprogramm zunächst mit einer simulierten Steuerung am Modell getestet werden (Sil). Dies hat gegenüber der Hil den Vorteil, dass noch nicht die gesamte Peripherie (Ein- und Ausgangsmodule) definiert und modelliert sein müssen. Denn, wird die reale Steuerung genutzt, ist es notwendig, die gesamte Peripherie und die Profibus-Verbindung ebenso zu simulieren, wie die gesamte Anlage durch das physikalische Modell.



Abbildung 5: Darstellung der Hil unter Verwendung einer realen Steuerung mit einem Human-Machine-Interface (HMI) und einer Simulation-Unit (1) und der SII mit einer simulierten Steuerung, simuliertem HMI (2) und dem multiphysikalischen Modell (3) im MCD, sowie beispielhafter, cloudbasierter Visualisierung von Produktionsdaten.

6 Anwendungsfälle

Die aufgezeigte Methodik des MBSE und eine Modellumgebung für eine Co-Simulation wurden anhand von zwei Anwendungsfällen im Bereich der Lehre und der Forschung umgesetzt. Die Ziele der beiden Fälle sind durchweg verschieden, sollen aber beide dazu beitragen, dass Produktentwicklung zukünftig, kostengünstiger, besser und effizienter durchgeführt werden kann.

6.1 Co-Simulation zur ganzheitlichen Simulation einer automatisierten Produktionsanlage

Die weiter anhaltende, oder besser gesagt, sukzessive ansteigende Verknappung von Ressourcen ist mittlerweile ein weltweites Problem, weshalb die Energiewende eingeleitet wurde. So sollen zukünftig im Kontext von Fertigungsanlagen erneuerbare Ressourcen wie Photovoltaikanlagen oder Windkraftwerke genutzt werden und Speichertechnologien (Akkumulatoren, Kondensatoren) in die Anlagen integriert werden. Es entstehen in den Fertigungshallen Smart Grids, die den Energiebedarf der Fertigung decken und die Verteilung der Ressourcen optimal nutzen. Die Energieversorgung muss deshalb über eine weitere Steuerung geregelt werden.

Das physikalische Verhalten von Akkumulator oder PV-Anlage lässt sich jedoch in den gängigen multiphysikalischen Simulationsumgebungen ebenso wenig abbilden, wie die Entnahme der Energie aus dem Versorgungsnetz. Hierzu müssen weitere Simulationen genutzt werden. Im vorliegenden Fall wurde für die Simulation des Akkumulators MATLAB Simulink genutzt. Für die Simulation des Prozessablaufs der Fertigungsanlage

wurde auf den MCD zurückgegriffen. In der vorliegenden Simulationsstudie wurden beide Umgebungen mittels eines OPC UA-Servers verbunden. Weiterhin verfügt jedes Tool über einen OPC UA-Client. Darüber werden in diskreten Zeitschritten Zustandsvariablen ausgetauscht und das Energiemanagement getestet werden. Beginnt beispielsweise in der Prozesssimulation ein energieintensiver Arbeitsschritt, so entlädt sich der Akkumulator in der eigenen Simulation maximal, wie in Abbildung 6 dargestellt ist.

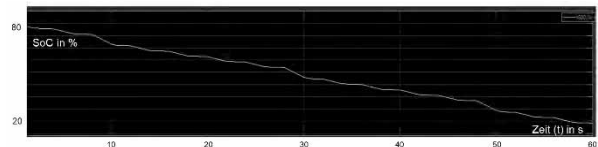


Abbildung 6: Verlauf des Ladezustandes über den Entladeprozess (in MATLAB Simulink)

Ist dieser vollständig entladen, wird der energiereiche Prozess gestoppt und erst wieder gestartet, wenn der Akkumulator durch die PV-Anlage geladen wurde.

Mit dieser Anwendung wird vor allem die Machbarkeit einer Co-Simulation mittels OPC UA bewiesen. Sämtliche Modelle sind aber im PLM-System abgelegt, denn der Aufbau des Produktentwicklungsprozesses wurde entlang des SE durchgeführt.

6.2 Praktikum für Studierende als Basis der zukünftigen Zusammenarbeit in Unternehmen

Ein weiterer Anwendungsfall diesbezüglich wurde in der Lehre umgesetzt. Die Studierenden erarbeiten in jedem Semester in einem interdisziplinären Team ein Entwicklungsprojekt anhand eines Fördersystems am lehrstuhligen I4.0-Demonstrator.

In einem fiktiven Entwicklungsteam werden verschiedene Rollen verteilt, deren Aufgabe es entweder ist, die mechanische, elektronische oder informationstechnische Konstruktion zu erstellen und zu kontrollieren, oder für die entwicklungsbegleitende Simulation Sorge zu tragen bzw. das Anlegen des Modells bereits in der Spezifikationsphase zu überwachen. Das Planspiel soll dazu beitragen, die beschriebenen Grundlagen des SE verstehen und nachvollziehen zu können, denn von der Aufnahme der Anforderungen aus den Erklärungen der Kunden bis zur Inbetriebnahme des Förderbandes durchlaufen die Studierenden alle Schritte des Produktentwicklungsprozesses, mit allen notwendigen Iterationen. Durch die Anwendung der Methodik des SE bereits im Studium, soll

ein Beitrag geleistet werden, Einführung und Anwendung in der Industrie voranzutreiben.

7 Zusammenfassung und Ausblick

Zusammenfassend ist festzustellen, dass der Ansatz, modellbasierter Entwicklung mithilfe von PLM-Systemen den Anforderungen, die durch Industrie 4.0 an Hersteller mechatronischer Systeme gestellt werden, gerecht werden kann. Hier spielt die Simulation eine wichtige Rolle. Simulation in einem multiphysikalischen Tool ist nicht ausreichend, so dass Co-Simulationen genutzt werden müssen. Einen Ansatz hierfür bietet die Verwendung des OPC UA-Standards. Dieser Ansatz kann erweitert werden, so dass damit eine virtuelle Inbetriebnahme durchgeführt werden kann. Um diesen Ansatz zu etablieren wird dieser möglichst praxisnah Studierenden vermittelt.

References

- [1] Armin Barnitzke. *Maschinensoftware muss modularer werden: Branchenumfrage*. In: Automationspraxis - die anwenderorientierte Fachzeitschrift für Führungskräfte in der Industrie; 2014 Dezember 2014; 12/2014. p. 1–3.
- [2] Harashima F, Tomizuka M, Fukuda T. *Mechatronics - "What Is It, Why, and How?" An editorial*. In: IEEE/ASME Transactions on Mechatronics 1996; 1: Nr. 1. p. 1–4.
- [3] 2206. 2004. *Richtlinie VDI 2206*. Entwicklungsmethodik für mechatronische Systeme
- [4] Renault O, Expert P IM. *Reuse/Variability Management and System Engineering*. In: Poster Workshop of the Complex Systems Design & Management Conference CSD&M 2014; 2014.
- [5] Kleiner S, Husung S, Schulze S O, Tschirner C, Kaffenberger R. *Model Based Systems Engineering: Prinzipien, Anwendung, Beispiele, Erfahrung und Nutzen aus Praxis*. In: Tag des Systems Engineering 2016. p. 13–22.
- [6] Eigner M, Roubanov D, Zafirov R. *Modellbasierte virtuelle Produktentwicklung*. Springer; 2014.
- [7] Fay Alexander. *Durchgängiges Engineering von Leitsystemen* (Konferenz für Durchgängige Anlagenplanung 2013). Nürnberg, 20.03.2013
- [8] Technical Operations International Council on Systems Engineering (INCOS). *Systems Engineering Vision (2025)*. URL https://www.incose.org/docs/default-source/about/se-vision-2025.pdf?sfvrsn=b69eb4c6_4 – Überprüfungsdatum 22.02.2021
- [9] Cooper R G. *Stage-gate systems: A new tool for managing new products*. In: Business Horizons 1990; 33: Nr. 3. p. 44–54.
- [10] Spath, D. *Produktionsarbeit der Zukunft - Industrie 4.0: Studie*. Stuttgart: Fraunhofer-Verl.; 2013.
- [11] Russwurm S. *Software: Die Zukunft der Industrie*. In: Sandler, U (Hrsg.): Industrie 4.0. Berlin, Heidelberg: Springer Berlin Heidelberg; 2013. p. 21–36.
- [12] Szinovatz A, Müller C. *Management der Komplexität im Innovationsprozess Vom Stage-Gate-Modell zum Survival-of-the-Fittest-Modell*. In: Schoeneberg, K-P (Hrsg.): Komplexitätsmanagement in Unternehmen. Wiesbaden: Springer Fachmedien Wiesbaden; 2014. p. 93–112.
- [13] Geisberger E, Schmidt R. *Abschlussbericht des Projekts "ProMiS" - Projektmanagement für interdisziplinäre Systementwicklungen: Aus dem Themenfeld "Software in technischen Produkten - Anwendung von Methoden und Vorgehensweisen für eine ingenieurmäßige Software-Entwicklung in der Produktion"* im Rahmen des Forschungsvorhabens des BMBF "Forschung für die Produktion von Morgen"; [Leitfaden für die Anwendung von Projektmanagement und Systemspezifikation mit einem Praxishandbuch auf CD-ROM] ; [Software VDMA] ; gefördert vom Bundesministerium für Bildung und Forschung. Frankfurt am Main, c 2004 (Software vfl77400)
- [14] Augustin C. *Leitfaden für die Anforderungsanalyse: Software aus der Reihe Methoden und Verfahren*. Frankfurt am Main: VDMA-Verl.
- [15] Sauer O, Schleipen M, Ammermann C. *Digitaler Fabrikbetrieb*. In: Zülch, G; Stock, P (Hrsg.): Integrationsaspekte der Simulation: Technik, Organisation und Personal. Karlsruhe: KIT Scientific; 2010. p. 559–566.
- [16] *VDI-Richtlinie 3633; Blatt 1*. 1993. VDI-Richtlinie 3633, Blatt 1
- [17] Feldmann K. *Simulationsbasierte Planungssysteme für Organisation und Produktion: Modellaufbau, Simulationsexperimente, Einsatzbeispiele*. Berlin [u.a.]: Springer; 2000.
- [18] Reinhart G, Feldmann K, Heitmann K. *Simulation-Schlüsseltechnologie der Zukunft*. In: Stand und Perspektiven. München: Utz, Wiss 1997.
- [19] Lechler T, Fischer E, Metzner M, Mayr A, Franke J. *Virtual Commissioning – Scientific review and exploratory use cases in advanced production systems*. In: Procedia CIRP 2019; 81. p. 1125–1130.
- [20] Siemens Industry Software GmbH & Co. KG. : *NX : Transformation des gesamten Produktentstehungsprozesses durch integrierte Softwarelösungen für Konstruktion, Simulation und Fertigung*. 2012
- [21] Günther F C: *Beitrag zur Co-Simulation in der Gesamtsystementwicklung des Kraftfahrzeugs*. München, TU München. 2017
- [22] Scheifele C: *Plattform zur Echtzeit-Co-Simulation für*

- die virtuelle Inbetriebnahme*. Stuttgart, Universität Stuttgart, Institut für Steuerungstechnik der Werkzeugmaschinen und Fertigungseinrichtungen: Dissertation. 2019
- [23] Pritschow G, Berkemer T, Bürger T, Croon N, Korajda B, Röck S. *Die simulierte Werkzeugmaschine*. In: Heisel, U (Hrsg.): Stuttgarter Impulse: Zukunft gestalten - Zeichen setzen ; FtK 2003, Fertigungstechnisches Kolloquium Stuttgart, 13. bis 15. Oktober 2003 ; Tagungsband ; [schriftliche Fassung der Vorträge. Stuttgart: Ges. für Fertigungstechnik; 2003. p. 219–246.
 - [24] Pritschow G, Röck S. *“Hardware in the Loop” Simulation of Machine Tools*. In: CIRP Annals 2004; 53: Nr. 1. p. 295–298.
 - [25] Schuh G. Innovationsmanagement: *Handbuch Produktion und Management 3. 2.*, vollst. neu bearb. und erw. Aufl. Berlin, Heidelberg: Springer; 2012.
 - [26] Eigner M, Stelzer R. *Product Lifecycle Management: Ein Leitfaden für Product Development und Life Cycle Management*. 2., neu bearb. Aufl. Dordrecht: Springer; 2013.
 - [27] Eigner M, Roubanov D, Zafirov R. *Modellbasierte virtuelle Produktentwicklung*. Berlin: Springer Vieweg; 2014.
 - [28] Sendler U. *Das PLM-Kompendium: Referenzbuch des Produkt-Lebenszyklus-Management*. Springer Science & Business Media; 2009.
 - [29] Lange J, Iwanitz F, Burke T J. *OPC: Von Data Access bis Unified Architecture*. 5., durchgesehene Auflage. Berlin, Offenbach: VDE Verlag GmbH; 2014.
 - [30] Mahnke W, Leitner S-H, Damm M. *OPC Unified Architecture*. 1st ed. Berlin: Springer; 2009.
 - [31] *VDI 4499*. 2008. VDI 4499 Blatt 1 Digitale Fabrik - Grundlagen
 - [32] WZL RWTHAachen. *PLM*. URL <http://www.plm-info.de> – Überprüfungsdatum 08.06.2015
 - [33] Wenk M. *Virtuelle Inbetriebnahme von Produktionsanlagen - Aufwand-Nutzen, Umsetzungsstrategien, zukünftige Entwicklungen*. In: VDE-Verlag (Hg.) 2008 – Elektrische Automatisierung - Systeme und Komponenten. p. 533.

Entwicklung einer Simulationsumgebung für die automatisierte Modellkonfiguration zur Auslegung und Absicherung KI-basierter Fahrfunktionen

Or Aviv Yarom^{*}, Xiaobo Liu-Henke

Institut für Mechatronik, Ostfalia Hochschule für angewandte Wissenschaften, Salzdahlumer Str. 46/48, 38302 Wolfenbüttel, Deutschland; ^{*}o.yarom@ostfalia.de

Abstract. Die Weiterentwicklung des autonomen Fahrbetriebs erfordert den vermehrten Einsatz innovativer und intelligenter Algorithmen. Um diese effektiv und effizient entwickeln zu können, sind geeignete Entwicklungsmethoden und -werkzeuge erforderlich. Deshalb wird in diesem Beitrag die Entwicklung einer Simulationsumgebung für die automatisierte Modellkonfiguration zur Auslegung und Absicherung KI-basierter Fahrfunktionen vorgestellt. Auf Basis des aktuellen Standes der Technik werden die Konzeption inklusive Anforderungsdefinition und Realisierung der Simulationsumgebung detailliert beschrieben. Zusätzlich wird die Simulationsumgebung in einer Anwendung zur automatisierten Fahrzeugführung mit Künstlichen Neuronalen Netzen validiert.

Einleitung

Der vom Europäischen Fonds für regionale Entwicklung (EFRE) geförderte Innovationsverbund *autoMoVe* (*Dynamisch konfigurierbare Fahrzeugkonzepte für den nutzungsspezifischen autonomen Fahrbetrieb*), hat die Entwicklung eines autonomen, modularen und elektrischen Fahrzeugkonzeptes zum Ziel. Durch den Austausch anwendungsspezifischer Module zur Laufzeit soll eine Vielzahl von Anwendungen vom innerbetrieblichen Gütertransport bis zur Personenbeförderung im Straßenverkehr autonom realisiert werden. Im Rahmen dieses Forschungsvorhabens fokussiert das Teilprojekt der Ostfalia *autoEVM* (*Ganzheitliches elektronisches Fahrzeugmanagement für autonome Elektrofahrzeuge*) die modellbasierte Entwicklung innovativer intelligenter Algorithmen und Funktionen für den autonomen Fahrbetrieb.

Mit einer höheren Automatisierung des Fahrbetriebs geht auch ein Anstieg der Anforderungen an das Fahrzeug bzw. die automatisierten Fahrfunktionen einher. Aktuelle Funktionen und Algorithmen, die auf Methoden der Regelungstheorie oder der klassischen Informationsverarbeitung basieren, können diesen nicht mehr in

vollem Umfang gerecht werden [1]. Deshalb stellt Künstliche Intelligenz (KI) in diesem Vorhaben bzw. für viele Domänen, die an der Entwicklung und Nutzung intelligenter, automatisierter Fahrzeuge beteiligt sind eine Schlüsseltechnologie dar [2].

Unabhängig von der Art der Informationsverarbeitung sind moderne Fahrzeuge und Fahrfunktionen komplexe mechatronische Systeme mit einem hohen internen und externen Vernetzungsgrad. Zur Beherrschung dieser Komplexität im Entwicklungs- und Absicherungsprozesses, wird auf eine in der Mechatronikforschung etablierte Entwurfsmethodik zurückgegriffen. Diese durchgängige und verifikationsorientierte Methodik basiert auf digitalen Modellen und Simulationen, um den Entwurfs- und Absicherungsprozess komplexer mechatronischer Systeme im vernetzten Umfeld einfacher, schneller und sicherer zu gestalten. [3]

Die zu diesem Zweck aktuell verfügbaren Entwicklungsumgebungen für intelligente Fahrzeugfunktionen sind zwar grundsätzlich sehr umfangreich, konzentrieren sich aber weitestgehend auf konventionelle Algorithmen zur Informationsverarbeitung. Der Einsatz von KI-Funktionen bei Entwicklung und Absicherung ist dabei nicht oder nur mit hohem Aufwand möglich. Umgekehrt bieten aktuelle Entwicklungsumgebungen für KI-Algorithmen nicht die Vorzüge bzw. den Funktionsumfang von Tools, die speziell auf automatisierte Fahrfunktionen ausgerichtet sind. [4]

Deshalb wird in diesem Beitrag die Entwicklung einer Simulationsumgebung für die automatisierte Modellkonfiguration zur Auslegung und Absicherung KI-basierter Fahrfunktionen vorgestellt. Diese Simulationsumgebung ist einerseits mit ihrem Funktionsumfang auf die Entwicklung automatisierter Fahrfunktionen ausgerichtet und bietet andererseits die Möglichkeit nicht nur konventionelle sondern auch KI-Algorithmen einzusetzen.

1 Entwurfsmethodik

Die Komplexität moderner Fahrzeuge steigt durch den höheren internen und externen Vernetzungsgrad sowie die wachsende Anzahl intelligenter und leistungsfähiger Hard- und Softwarekomponenten stetig an. Zur Beherrschung der Systemkomplexität und frühzeitigen Fehlervermeidung bei der Auslegung der Informationsverarbeitung, ist eine ganzheitliche Entwurfsmethodik unverzichtbar. Daher hat sich die durchgängige, verifikationsorientierte, modellbasierte Entwurfsmethodik, basierend auf dem Rapid Control Prototyping (RCP) und Model-in-the-Loop-(MiL), Software-in-the-Loop-(SiL) und Hardware-in-the-Loop-(HiL) Simulationen etabliert. [3]

Die Methodik baut auf funktionsorientierten physikalischen Modellen einer Regelstrecke auf. Die Regelungsfunktion wird anschließend in Abhängigkeit des Systemverhaltens simulativ ausgelegt und frühzeitig in MiL-Simulationen validiert. Um manuelle Programmierung zu umgehen, werden Modell und Regelungsfunktion in blockschaltbildbasierten Programmiersprachen entwickelt. Der anschließend automatisch generierte Funktionscode wird in SiL-Simulationen erneut gegen das Streckenmodell getestet. HiL-Simulationen dienen zur weiteren Validierung und Optimierung der Informationsverarbeitung mit echtzeitfähigen Simulationsmodellen und realen Teilkomponenten des zu regelnden Systems.

Der verifikationsorientierte und iterative Ansatz dieser Methodik unterstützt den Entwicklungsprozess auch in der herausfordernden Aufgabe der Absicherung. Die Methodik adressiert die Schwächen der klassischen auf physischen Prototypen basierenden Validierung, wie einem hohem Ressourcenaufwand oder Sicherheitsrisiken für Mensch, Maschine und Umwelt. Durch ihren virtuellen Charakter sparen MiL-, SiL- und HiL-Simulationen Zeit und Kosten [5]. Sie ermöglichen jederzeit durchführbare und reproduzierbare Tests ohne direkte Abhängigkeit von physischen Prototypen, Tageszeiten oder menschlichen Experten. So lassen sich Simulationsdurchläufe, für verschiedene Funktionsvarianten oder Szenarien automatisiert durchführen. Damit eignet sich diese Methodik auch besonders für das Training von KI-Algorithmen. Denn das sogenannte maschinelle Lernen verläuft, bis auf seltene Ausnahmen, immer iterativ.

Virtuelle Entwurfsmethoden wie diese bilden die Grundlage für viele intelligente Systeme, wie hochautomatisierte Fahrzeuge. Mit prototypenbasierten Tests wären die benötigten Hunderttausenden Testkilometer nicht mit vertretbarem Zeit- und Kostenaufwand zu leisten. [5]

2 Stand der Technik

2.1 Intelligente Fahrfunktionen

Beim automatisierten Fahren werden dem menschlichen Fahrer einzelne Fahraufgaben von sogenannten Fahrerassistenzsystemen (FAS) abgenommen. Solche FAS, z.B. zur Geschwindigkeitsregelung oder Spurhalteassistenten sind schon seit längerem in Serie verfügbar [6]. FAS verarbeiten von Fahrzeug- und Umgebungssensoren erfasste Daten und berechnen dadurch Fahrbefehle, welche anschließend mittels geregelter Aktorik umgesetzt werden. Mit steigender Anzahl und Vernetzung der FAS delegiert der menschliche Fahrer sukzessive Fahraufgaben an das Fahrzeug bis er beim autonomen Fahrbetrieb schließlich zum Passagier wird. Dann spricht man nicht mehr von FAS, sondern von (automatisierten) Fahrfunktionen.

Eine ansteigende Automatisierung des Fahrbetriebs bedeutet eine extreme Steigerung der Anforderungen an die Fahrfunktionen. So werden nicht nur mehr, sondern auch verschiedene Sensoren zur Umfeldperzeption benötigt, deren inhomogene Rohdaten mehrfach, intensiv und mit höchsten Echtzeitanforderungen verarbeitet und umgesetzt werden müssen. Kommen Informationen aus der internen Bus- oder externen Vehicle-to-Everything-(V2X-) Kommunikation hinzu, steigt die Komplexität weiter an. Dadurch geraten konventionelle Algorithmen zur Regelung und Datenverarbeitung an ihre Grenzen. [1] Deshalb werden bereits heute KI-Algorithmen für automatisierte Fahrfunktionen eingesetzt. Besonders vielversprechend sind dabei Künstliche Neuronale Netze (KNN) in Kombination mit dem Maschinellen Lernen (ML). Prominente Anwendungen sind die bildbasierte semantische Segmentierung der Fahrumgebung [7] oder die automatisierte Fahrzeugführung [8]. KI-Algorithmen sind aufgrund ihrer Leistungsfähigkeit, Robustheit und Anpassungsfähigkeit vielversprechend für die Weiterentwicklung autonomer Fahrzeuge [9]. Dabei bietet sich auch außerhalb der automatisierten Fahrzeugführung Potential für intelligente Fahrzeugfunktionen, z.B. für das Batteriemanagement [10]. Auf niedrigen Ebenen der Informationsverarbeitung z.B. der lokalen Regelung der Aktorik werden hingegen häufig weiterhin konventionelle Ansätze verwendet [1].

2.2 Grundlagen Künstlicher Neuronaler Netze und Maschinellen Lernens

Der Begriff KI umfasst eine Vielzahl unterschiedlichster Methoden und Algorithmen, die sich mit der

selbstständigen und automatisierten Lösung von Problemen befassen [2]. KNN und ML bilden ein Teilgebiet der KI, das sich in zahlreichen Problemen verschiedenster Gebiete, auch beim autonomen Fahrbetrieb, als geeignet erwiesen hat. Deshalb fokussiert sich dieser Beitrag auf dieses Teilgebiet. Die zahlreichen positiven Eigenschaften von KNN und ML, wie Anpassungsfähigkeit, Fehlerresistenz, Vielseitigkeit und vor allem Lernfähigkeit sind auf ihre Anlehnung an die Struktur und Funktionsweise des menschlichen Gehirns zurückzuführen.

Analog zur Biologie sind (künstliche) Neuronen Verarbeitungseinheiten, die Eingangsreize über gewichtete Verbindungen kumulieren und mit Hilfe einer Aktivierungsfunktion eine Ausgabe berechnen. Durch die Zusammenschaltung mehrerer Neuronen in mindestens zwei Schichten entsteht das KNN. Üblich sind Kombinationen von bis zu einigen Hundert Neuronen in bis zu über Einhundert Schichten. Dabei sind nicht nur beliebige vorwärtsgerichtete, sondern auch zeitlich rückgekoppelte Verbindungen im KNN möglich. Die optimale Architektur eines KNN lässt sich bisher nicht analytisch bestimmen [11]. Deshalb sind Erfahrung und Versuchsreihen notwendig, um eine geeignete Architektur im Zielkonflikt zwischen Rechenaufwand und Leistungsfähigkeit zu finden. Die Anzahl, Verschaltung und Gewichtung der Verbindungen charakterisiert die „Intelligenz“ eines KNN. Allgemein gesagt, bedeuten mehr Neuronen und Verbindungen eine höhere Leistungsfähigkeit des KNN, bei gleichzeitig steigendem Rechenaufwand.

Genau wie ein menschliches Gehirn, muss das KNN eine Aufgabe zunächst lernen bzw. trainieren. Diese Begriffe bezeichnen die Anpassung der Verbindungsgewichte. Im Umfeld des autonomen Fahrens sind dafür das Supervised Learning (SL) und das Reinforcement Learning (RL) relevant. Beim SL werden dem KNN Eingangsdaten und die zugehörige Ausgabe präsentiert. Das KNN lernt iterativ den Zusammenhang zwischen den beiden Größen [12]. Dieses Lernverfahren eignet sich z.B. besonders für bildbasierte Objekterkennung [7]. Beim RL lernt das KNN sukzessive aus der Erfahrung vergangener Sequenzen die optimale Strategie im Sinne einer vorgegeben Belohnungsfunktion [12]. Dieses Verfahren wird eingesetzt, wenn keine Trainingsdaten verfügbar sind, z.B. bei der automatisierten Fahrzeugführung [8]. SL und RL sind Oberkategorien von Lernverfahren, mit diversen konkreten Trainingsalgorithmen. Genau wie die KNN Architektur lassen sich die optimalen Trainingsalgorithmen bzw. deren Parameter nicht analytisch bestimmen. So sind auch hier Erfahrung und Versuche erforderlich.

2.3 Entwicklungsumgebungen für intelligente Fahrfunktionen

Für den modellbasierten Entwurf automatisierter Fahrfunktionen existieren einige Entwicklungs- und Simulationsumgebungen, wie MATLAB/Simulink mit Toolboxen, dSPACE Automotive Simulation Models (ASM), und IPG CarMaker, um nur einige zu nennen. Sie alle unterscheiden sich hinsichtlich ihres primären Fokus oder ihrer spezifischen Vor- und Nachteile. Alle bieten umfangreiche Modellbibliotheken für Verkehrs-, Fahrdynamik-Komponenten-, Sensor- oder sogar Fahrermodelle. Sie wurden speziell für die Konfiguration und reproduzierbare Simulation verschiedenster Fahrscenarien und Fahrzeugvariationen zum Zwecke der Fahrzeugentwicklung oder -absicherung entwickelt. Die Tools besitzen meist integrierte Modell-Konfiguratoren, eine Visualisierung, Experimentierumgebungen, teilweise auch ein Szenario- und Testmanagement. Die aufgezählten Tools eignen sich für die in Kapitel 1 beschriebene Entwurfsmethodik mittels RCP-Prozess. Sie ermöglichen nicht nur MiL-Simulationen, sondern durch automatische Codegenerierung in Verbindung mit MATLAB/Simulink auch echtzeitfähige SiL- und HiL-Simulationen. [13] Allerdings bieten die Tools keine Möglichkeit ohne weiteres KNN in die jeweiligen Simulations- und Konfigurations-tools einzubinden, geschweige denn diese zu trainieren.

Die einzige Ausnahme stellt hier die Deep Learning Toolbox (DLT) von MATLAB dar. Mit dieser Toolbox können beliebige KNN Architekturen konfiguriert und mit einer Vielzahl vorimplementierter Algorithmen performant trainiert werden. Die DLT ist grundsätzlich kompatibel mit Simulink sowie der automatischen Codegenerierung. Die KNN Konfiguration ist aber recht kompliziert und erfolgt in der Regel manuell für einzelne KNN. Dadurch eignet sie sich in ihrer Ursprungsform wenig für die Durchführung von Versuchsreihen mit variierenden Architekturen, Trainingsparametern oder auch unterschiedlichen Fahrfunktionen.

Auch für das Training von KNN existieren spezielle Entwicklungsumgebungen. Die meisten von ihnen basieren auf der Programmiersprache Python. Prominente Tools sind unter anderem TensorFlow, PyTorch, Keras und Caffe. Sie alle besitzen komfortable und umfangreiche Funktionalitäten hinsichtlich der Erstellung und des Trainings von KNN. Allerdings gibt es hier nur die Möglichkeit zur skriptbasierten Programmierung. Eine übersichtlichere für den RCP-Prozess geeignete blockschaltbildbasierte Programmierung ist nicht möglich. [4] Daher

gestaltet es sich schwer mit den KNN-spezifischen Tools gleichwertige Simulationen hinsichtlich Genauigkeit, Rechenzeit und Komfort zu entwerfen, wie mit den Tools für automatisierte Fahrfunktionen. Der Aufwand hierfür ist so groß, dass sogar Computerspiele wie Grand Theft Auto [14] oder TORCS [15] für Fahrzeug- und Verkehrssimulationen in Kombination mit den genannten KNN-Tools gekoppelt wurden. Diese Ansätze dienten jedoch eher der Untersuchung der KNN und des ML und werden den Ansprüchen an Genauigkeit, Echtzeitfähigkeit, Reproduzierbarkeit und Variationsmöglichkeit einer realen Fahrfunktionsentwicklung nicht gerecht.

3 Konzeption der Simulationsumgebung

3.1 Ableitung der Problemstellung

Mit den Erkenntnissen aus Kapitel 2 lässt sich eine Problemstellung für die Verfolgung der Projektziele (Kapitel 1) ableiten. Der Entwurf intelligenter automatisierter Fahrfunktionen erfordert den Einsatz von KI-Algorithmen. Diese lassen sich aber, wie beschrieben nicht ohne weiteres in den üblichen Entwicklungsumgebungen einsetzen. KNN und ML Verfahren werden sehr schnell sehr komplex und unübersichtlich. Eine händische Programmierung verschiedener KNN-Architekturen mit den zugehörigen Berechnungsvorschriften wäre nicht nur fehleranfällig, sondern würde auch viel Zeit in Anspruch nehmen. Analog verhält es sich mit den ML-Verfahren, die parametrisiert und für jede KNN-Architektur angepasst werden müssten. Der Umstand, dass Architekturdesign, Trainings- und Testprozesse stets empirisch und iterativ ablaufen, erfordert viele Simulationsdurchläufe und verschlimmert die Situation weiter.

Der Einsatz von KNN-spezifischen Entwicklungsumgebungen ist ebenfalls ausgeschlossen. Will man die automatisierten Fahrfunktionen unter realistischen Bedingungen auslegen, muss die Modellierungstiefe ausreichend exakt sein. Beim klassischen Weg der analytischen physikalischen Modellbildung müssten die mathematischen Gleichungen und deren numerische Lösungsverfahren selbstständig in einer Skriptsprache programmiert werden. Betrachtet man nun die Anzahl der Subsysteme eines Fahrzeugs, weitere Fahrfunktionen oder andere Verkehrsteilnehmer, die mitsimuliert werden sollen, ergibt sich ein hochkomplexes Simulationssystem. Hinzu kommen verschiedene Varianten, Konfigurationen,

Szenarien und gegebenenfalls Echtzeitanforderungen. Die Umsetzung eines solchen Vorhabens wäre zwar grundsätzlich möglich, aber sehr fehleranfällig und wenig zielführend. Schließlich gibt es bereits ressourcenoptimale Simulationsumgebungen für diesen Zweck.

Zusammenfassend eignet sich also keine der derzeitigen verfügbaren Simulationsumgebungen für die Auslegung und Absicherung KI-basierter Fahrfunktionen nach der Entwicklungsmethodik aus Kapitel 1. Als Schlussfolgerung ergibt sich demnach, die Entwicklung einer eigenen Simulationsumgebung für diese Entwurfsmethodik.

3.2 Definition von Anforderungen

Zur Bewältigung der genannten Herausforderungen, werden im Folgenden die Anforderungen an die neue Simulationsumgebung für die automatisierte Modellkonfiguration zur Auslegung und Absicherung KI-basierter Fahrfunktionen definiert:

- A1. Nutzbar für verschiedene Fahrfunktionen
- A2. Betrieb mit Simulink und Kompatibilität mit entsprechenden Blöcken und Modellen
- A3. Nutzung verschiedenster vorhandener oder Erstellung eigener Modelle und Funktionen mit beliebiger Modellierungstiefe
- A4. Einfache Erstellung und Berechnung von KNN
- A5. Automatische Integration von KNN in die Simulinkmodelle
- A6. Variation der Modellkonfiguration
- A7. Automatische Modellkonfiguration
- A8. Automatisierter Durchlauf von Simulationsreihen
- A9. Training mittels verschiedener ML Verfahren
- A10. Einsatz zur Generierung von Trainingsdaten für ML
- A11. Visualisierung in 2D und 3D
- A12. Bedienung mit Nutzeroberfläche oder skriptbasiert
- A13. Performante Rechenzeiten für große Simulationsreihen und spätere Echtzeitanwendungen
- A14. Kompatibilität mit dSPACE ASM für die spätere Weiterentwicklung der Simulationsumgebung
- A15. Automatische Codegenerierung möglich

3.3 Konzeptbildung

Der erste Schritt zur Erfüllung der zuvor definierten Anforderungen, ist die Realisierung der Simulationsumgebung in MATLAB und Simulink (A2). Dadurch ist die Kompatibilität zu Standardblöcken und bereits vorhandenen Modellen, die in Simulink verfügbar sind, gegeben (A3). Des Weiteren können beliebige weitere Modelle und automatisierte Fahrfunktionen erstellt werden (A1).

Hierbei kann grundsätzlich auch auf Toolboxen zurückgegriffen werden. Dabei ist lediglich darauf zu achten, dass diese auch die automatische Codegenerierung unterstützen (A15). In diesem Fall ist auch die Kompatibilität zu dSPACE ASM gewährleistet (A14).

Für die Konfiguration und Erstellung von KNN wird auf die DLT zurückgegriffen. Hierfür wird ein zusätzlicher KNN Generator entworfen, der beliebige KNN-Architekturen durch einfache Benutzereingaben über eine grafische Benutzeroberfläche (GUI) oder ein Skript mit einem automatisiert erstellt (A4) und direkt in das Simulationsmodell generiert (A5). Durch die Verwendung der DLT sind automatisch verschiedene ML Algorithmen verfügbar (A9). Für das SL sind bereits mehrere Algorithmen nutzbar, sofern man Trainingsdaten generiert hat (A10). RL Algorithmen sind in der RL Toolbox von Matlab verfügbar. Alternativ ist es sowohl für SL als auch für RL grundsätzlich auch möglich eigene Trainingsalgorithmen zu implementieren, da man mit der DTL recht komfortabel auf die Verbindungsgewichte zugreifen kann.

Mit einem Szenario-Generator können verschiedene Simulationsszenarien hinsichtlich der Fahrstrecke und der Verkehrsteilnehmer konfiguriert werden (A6). Diese Konfiguration wird anschließend automatisch an das Simulationsmodell übergeben. Um das Geschehen im Szenario während und nach der Simulation besser interpretieren zu können, wird zusätzlich eine rudimentäre Visualisierung implementiert. Die 2D- bzw. 3D-Visualisierung basiert auf dem Bird's-Eye Scope und der Plot-Funktion von MATLAB (A11).

Die Konfiguration und Erstellung des Szenarios erfolgt ebenfalls durch eine GUI oder ein Skript (A12). Durch die Möglichkeit der skriptbasierten Szenario- und KNN-Generierung, lassen sich Simulationsabläufe automatisieren (A8). Dabei werden die entsprechenden Modellparameter automatisch konfiguriert und aktualisiert (A7). Genau wie bei den entwickelten Fahrfunktionen und Modellen, muss auch bei der Realisierung von Szenario- und KNN-Generator sowie den Automatisierungsmechanismen auf eine schlanke und rechenzeitoptimierte Programmierung geachtet werden (A13).

4 Realisierung der Simulationsumgebung

Abbildung 1 zeigt Aufbau und Struktur der Simulationsumgebung. Sie besteht im Wesentlichen aus dem Simulationsmodell selbst und aus der Steuerung der

Simulationsumgebung. Im Simulationsmodell finden die Modellierung und die Berechnung der Simulation statt. Darin enthalten sind sowohl die Modelle, Parameter und Funktionen des gesamten Ego-Fahrzeugs in der gewünschten Modellierungstiefe als auch die im Szenario definierte Umgebungssimulation. Die zu entwickelnde automatisierte Fahrfunktion, also beispielsweise das KNN ist im Punkt Fahrfunktion zusammengefasst. Sie interagiert mit den anderen simulierten Bestandteilen, wie den Sensor-, Fahrdynamik-, Komponenten- oder Kommunikationsmodellen. Gleichmaßen kommunizieren die Modellbestandteile über definierte Schnittstellen mit den Komponenten des Simulationsszenarios.

Bevor eine Simulation stattfinden kann, muss zunächst eine Konfiguration der Simulationen stattfinden. So teilt sich die Steuerung der Simulationsumgebung in zum einen in die Konfiguration der Simulationsdurchläufe, zum anderen in die Simulationssteuerung und

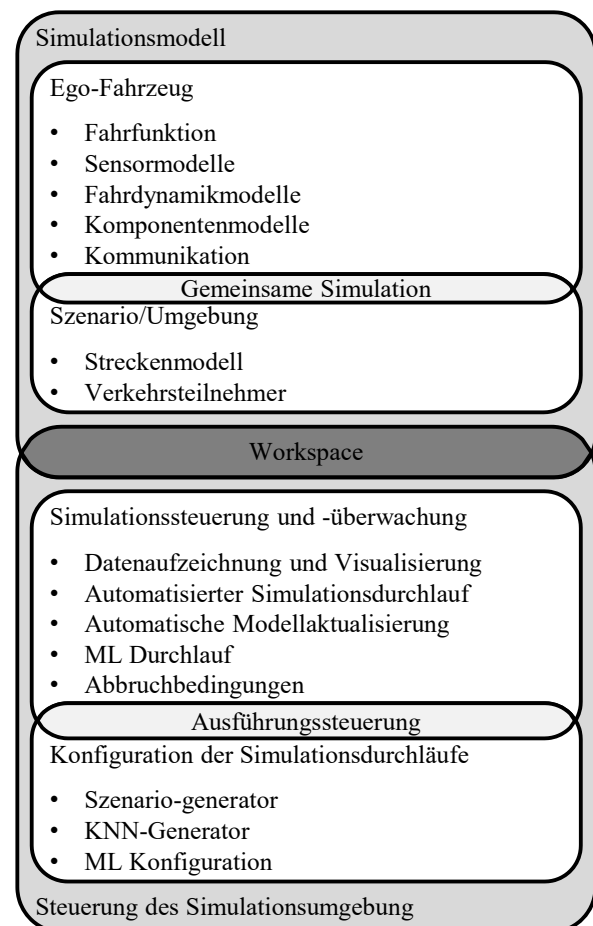


Abbildung 1: Aufbau und Struktur der Simulationsumgebung für die automatisierte Modellkonfiguration zur Auslegung und Absicherung KI-basierter Fahrfunktionen

Überwachung. Die Konfiguration der Simulationsdurchläufe ist im Grunde die Eingabeschnittstelle des Nutzers. Mit dem Szenario-Generator kann entsprechend der Funktionsspezifikation ausgewählt werden wie genau jede einzelne Simulation aussehen und ablaufen soll. Einstellbar sind z.B. der Streckenverlauf, die Art, Anzahl und Bewegung anderer Verkehrsteilnehmer usw. Der KNN-Generator ermöglicht die Auswahl der KNN-Architekturen für die verschiedenen Durchläufe. In der ML Konfiguration werden Lernalgorithmen ausgewählt und parametrisiert. Die Informationen aus der Nutzereingabe dienen zur Ausführungssteuerung für die Simulationssteuerung und -überwachung.

Es wird also ein Ablaufplan zur Automatisierung der Simulationsdurchläufe erzeugt. Darin werden z. B. mehrere ML Durchläufe zunächst in einem und dann in einem weiteren Szenario gestartet. Vor jedem Durchlauf werden die entsprechenden Parameter der jeweils aktiven Konfiguration geladen und über den MATLAB Workspace an das Simulationsmodell übergeben. So wird das Modell automatisch aktualisiert. Der Workspace dient als Schnittstelle zwischen Steuerung der Simulationsumgebung und dem Simulationsmodell. Er ermöglicht die Aufzeichnung und Speicherung der Simulationsdaten für die Visualisierung und spätere Auswertung. Des Weiteren überträgt er ausgewählte Größen, zur Modellüberwachung. Bei vorher definierten, unzulässigen Zuständen, wird die die Simulation automatisch abgebrochen.

5 Simulation und Auswertung

5.1 Beschreibung des Anwendungsfalls und der Modellbildung

Zur Veranschaulichung soll in diesem Anwendungsfall eine Fahrfunktion zur automatisierten Querführung bei konstanten Geschwindigkeiten auf beliebigen, einspurigen Strecken ohne andere Verkehrsteilnehmer ausgelegt werden. Dabei ist das Beispiel bewusst komprimiert gewählt, um zu zeigen welcher Simulations- und Zeitaufwand entsteht. Dadurch soll der Mehrwert der automatisierten Modellkonfiguration verdeutlicht werden.

Da in diesem Anwendungsfall die Fahrzeuggeschwindigkeit konstant ist, wird ein lineares Einspurmodell zur Abbildung der Fahrdynamik eingesetzt. Automatisch generierte Strecken nach Richtlinien der Bundesanstalt für Straßenwesen mit einer Gesamtbreite von 3,5 m bilden die Umgebung. Das Sensormodell besteht aus elf Linien, die in einem Winkelbereich von $\pm 40^\circ$ und einem

Radius von 8 m die Entfernung zu den Fahrbahnbegrenzungen erkennen. Abbildung 2 a) und b) veranschaulichen den Anwendungsfall mit der Visualisierungsfunktion der Simulationsumgebung. Andere Fahrzeugkomponenten und Kommunikationssysteme werden nicht berücksichtigt. Die automatisierte Querführung wird von einem KNN ausgeführt. Dabei sind die elf Sensorwerte die Eingänge, ein zugehöriger Lenkwinkel der Ausgang.

Das KNN soll mit einem selbst implementierten RL-Verfahren, den sogenannten Genetischen Algorithmen (GA) ein natürliches Lenkverhalten erlernen. GA sind eine Gruppe von Algorithmen, die den natürlichen Prozess der Evolution nachahmen, um sich sukzessive einer optimalen Lösung anzunähern. Dabei entwickelt sich eine sogenannte Population, die aus mehreren Lösungskandidaten (Individuen) besteht, generationsweise durch Selektion, Rekombination und Mutation weiter. Eine Populationsgröße von 50 bedeutet hier, dass 50 verschiedene KNN in einer Generation simuliert werden. Der iterative Charakter des GA resultiert in tendenziell relativ vielen Simulationen.

In Abbildung 2 c) ist eine Übersicht über den grundlegenden Aufbau des Simulationsmodells dargestellt. Der GA befindet sich eigentlich außerhalb des Simulationsmodells. Die zugehörige Belohnungsfunktion greift auf Simulationsdaten zu, um das Verhalten der Individuen zu bewerten. Diese Information nutzt der GA, um die Verbindungsgewichte des KNN zu aktualisieren.

5.2 Konfiguration der Simulationsdurchläufe

Wie in Kapitel 2 beschrieben, verläuft die Bestimmung der KNN-Architektur und der Parameter des ML-Verfahrens immer empirisch. Darüber hinaus ist es wichtig beim

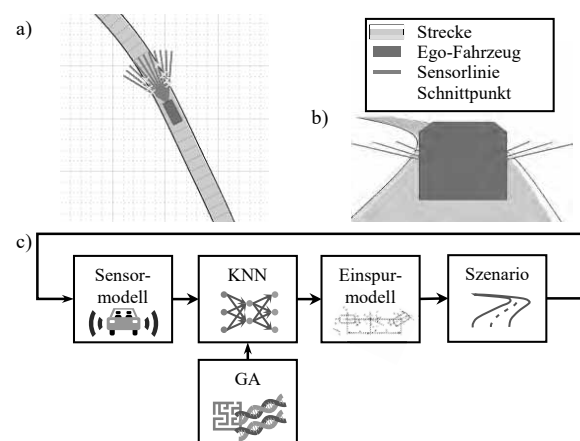


Abbildung 2: Überblick über den Anwendungsfall und die Modellbildung

Training von KNN auf die Generalisierungsfähigkeit zu achten. Das bedeutet in diesem Fall, dass das KNN in der Lage sein muss, auch auf unbekannten Strecken die automatisierte Querführung auszuführen. Um diesen Aspekt direkt während des Trainings zu berücksichtigen, werden mehrere Testdurchläufe direkt nach einzelnen Trainings durchgeführt. Es ist also erforderlich mehrere verschiedene Simulationen zu konfigurieren.

Um sukzessive zu einer ideal ausgelegten Fahrfunktion zu gelangen, müssen mehrstufige Versuche in der Konfiguration der Simulationsdurchläufe implementiert werden. Dabei besteht der Simulationsdurchlauf immer aus mehreren ML Durchläufen. Ein ML Durchlauf beinhaltet immer ein Training einer KNN-Architektur mit einem GA-Parametersatz auf einer Strecke mit einer Belohnungsfunktion, gefolgt von vier Testdurchläufen auf weiteren Strecken. Während die Trainingsphase des GA aus vielen einzelnen Simulationen für die Individuen und Generationen besteht, wird bei den Testdurchläufen jeweils nur das beste Individuum aus dem Training geprüft. Die Anzahl der durchzuführenden Simulationen S_i pro ML Durchlauf i ergibt sich also in Abhängigkeit der Populationsgröße P_i , der Generationenzahl G_i und Anzahl der Testdurchläufe T_i . Die Anzahl der Simulationen im Gesamtdurchlauf N_G ist die Summe aller S_i für die verschiedenen ML Durchläufe bzw. -Konfigurationen K_{ML} :

$$N_G = \sum_{i \in K_{ML}} S_i = \sum_{i \in K_{ML}} P_i \cdot G_i + T \quad (1)$$

Bei einer angenommenen Populationsgröße von 50 und einer Generationenzahl von 25, werden in einem ML Durchlauf folglich 1254 Simulationen durchgeführt.

Die erste Konfiguration sieht die Untersuchung von sechs Varianten der GA-Parametersätze für das Training eines Basis-KNN vor (Abbildung 3 a)). Das Basis-KNN ist ein nicht optimiertes KNN von dem man ausgeht, dass es die Funktion grundlegend erfüllen kann. Analog verhält es sich mit der Basis-Belohnungsfunktion. Der so ermittelte optimale GA-Parametersatz wird, wie in Abbildung 3 b) gezeigt, genutzt, um die beste aus zwölf vorkonfigurierten KNN-Architekturen zu bestimmen. Im letzten Schritt (Abbildung 3 c)) erfolgt die eigentliche Optimierung des KNN-Verhaltens durch weitere ML Durchläufe mit neun verschiedenen Belohnungsfunktionen. Die Anzahl der insgesamt durchgeführten Simulationen N_G liegt nach Gleichung (1) im hohen sechsstelligen Bereich. Ohne die automatische Modellkonfiguration der automatisierten Simulationsumgebung, hätte der benötigte Aufwand das vertretbare Maß überschritten.

5.3 Simulationsergebnisse und Auswertung

Nach dem automatischen Durchlauf aller Konfigurationen aus Abbildung 3, ist eine Funktion zur automatisierten Querführung entstanden. Das KNN kann auf beliebigen Strecken in einem Geschwindigkeitsbereich von 30 bis 70 km/h sicher und komfortabel die Querführung übernehmen. Für detailliertere Beschreibungen der Auslegung und Absicherung der Fahrfunktion aus diesem Anwendungsfall, sei auf die Vorarbeiten [5] und [16] verwiesen. Der Schwerpunkt dieses Beitrags liegt auf der Simulationsumgebung. Zu deren Auswertung wurde ein einzelner automatisierter ML-Durchlauf händisch in MATLAB programmiert und hinsichtlich der Laufzeit mit der Simulationsumgebung verglichen. Bei der händisch programmierten Version wurden auch alle Modellbestandteile und die Visualisierung selbst implementiert.

Der Vergleich zeigte, dass ein ML Durchlauf in der händisch programmierten Umgebung etwa 22 h und 36 min dauert. In der Simulationsumgebung für die automatisierte Modellkonfiguration hingegen, dauert ein ML Durchlauf mit identischer Konfiguration mit ca. 28 min nur etwa 2 % dieser Zeit. Weiterhin ist es möglich, die Visualisierung in der Simulationsumgebung abzuschalten. So lässt sich die benötigte Dauer je ML Durchlauf noch einmal vierteln.

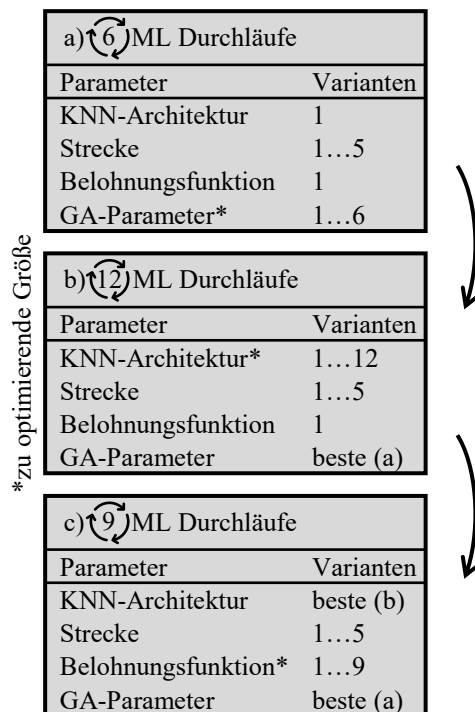


Abbildung 3: Ablauf des automatisierten Simulationsdurchlaufs

6 Zusammenfassung und Ausblick

In diesem Beitrag wurde eine Simulationsumgebung für die automatisierte Modellkonfiguration zur Auslegung und Absicherung KI-basierter Fahrfunktionen vorgestellt. Ausgehend von einer Einleitung und Motivation wurden die Entwurfsmethodik und der Stand der Technik vorgestellt. Hieraus wurde die Notwendigkeit zur Entwicklung der vorgestellten Simulationsumgebung abgeleitet und Anforderungen für diese definiert. Die Umsetzung der Anforderungen wurde im Konzept und der Realisierung beschrieben. Das Ergebnis ist eine Simulationsumgebung auf Basis von MATLAB und Simulink, die einen hohen Kompatibilitätsgrad zu vorhandenen Entwicklungsumgebungen aufweist. Außerdem kann sie große Simulationsreihen nach einer benutzerfreundlichen Konfiguration in kurzer Zeit automatisch durchführen und visualisieren. Damit eignet sie sich hervorragend für den Einsatz in der vorgestellten Entwurfsmethodik. Schließlich wurde die Simulationsumgebung in einem Anwendungsbeispiel eingesetzt und so deren Mehrwert und Funktionsfähigkeit nachgewiesen. Zukünftige Arbeitsschritte umfassen unter anderem die Erweiterung der Modell- und Funktionsbibliothek sowie die Integration mit dSPACE ASM.

Danksagung

Diese Veröffentlichung entstand aus dem Teilprojekt "autoEVM" (*Ganzheitliches elektronisches Fahrzeugmanagement für autonome Elektrofahrzeuge*) (ZW 6-85030889), im Rahmen des Innovationsverbundes "auto-MoVe" (*Dynamisch konfigurierbare Fahrzeugkonzepte für den nutzungsspezifischen autonomen Fahrbetrieb*), welcher vom Europäischen Fonds für regionale Entwicklung (EFRE) gefördert und vom Projektträger NBank verwaltet wird.



Literatur

- [1] Milz S., Schrepfer J. Is artificial intelligence the solution to all our problems? Exploring the applications of AI for automated driving. In: Bertram T. (eds) *Automatisiertes Fahren 2019*. Springer Vieweg, Wiesbaden, 2020.
- [2] Schiekofner, P. et al. Maschinelles Lernen für das automatisierte Fahren. *ATZ Automobiltech Z*, vol. 121, 2019.
- [3] Liu-Henke X. et. al. Holistic development of a full-active electric vehicle by means of a model-based systems engineering. *2016 IEEE International Symposium on Systems Engineering (ISSE)*, Edinburgh, UK, 2016.
- [4] Stančin I., Jović A. An overview and comparison of free Python libraries for data mining and big data analysis. *2019 42nd International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)*, Opatija, Croatia, 2019.
- [5] Yarom O. A. et. al. Artificial Neural Networks and Reinforcement Learning for model-based design of an automated vehicle guidance system. *12th International Conference on Agents and Artificial Intelligence (ICAART)*, Valletta, Malta, 2020.
- [6] Reif K. Fahrerassistenzsysteme. In: *Automobilelektronik*. Vieweg+Teubner Verlag, 2012.
- [7] Lyu H. et. al. Esnet: Edge-Based Segmentation Network for Real-Time Semantic Segmentation in Traffic Scenes. *2019 IEEE International Conference on Image Processing (ICIP)*, Taipei, Taiwan, 2019.
- [8] Huang Z. et. al. Parameterized batch reinforcement learning for longitudinal control of autonomous land vehicles. *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 49, no. 4, 2019.
- [9] Kuutti S. et. al. A Survey of Deep Learning Applications to Autonomous Vehicle Control, *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 2, 2021.
- [10] Alaoui C. Hybrid Vehicle Energy Management Using Deep Learning. *2019 International Conference on Intelligent Systems and Advanced Computing Sciences (ISACS)*, Taza, Marokko, 2019.
- [11] Tirumala S.S. Evolving deep neural networks using co-evolutionary algorithms with multi-population strategy. In: *Neural Comput & Applic*, vol. 32, 2020.
- [12] Duriez T., Brunton S., Noack B. R. *Machine Learning Control*. Springer International Publishing, Cham, Switzerland, 2017.
- [13] Deter D. et. al. Simulating the Autonomous Future: A Look at Virtual Vehicle Environments and How to Validate Simulation Using Public Data Sets. in *IEEE Signal Processing Magazine*, vol. 38, no. 1, 2021.
- [14] Wang D. et. al. Deep object centric policies for autonomous driving. arXiv preprint arXiv:1811.05432, 2018.
- [15] Zhang J., Cho K. Query-efficient imitation learning for end-to-end autonomous driving,” arXiv preprint arXiv:1605.06450, 2016.
- [16] Yarom O. A., Jacobitz S., Liu-Henke X. Design of Genetic Algorithms for the Simulation-Based Training of Artificial Neural Networks in the Context of Automated Vehicle Guidance. *2020 19th International Conference on Mechatronics - Mechatronika (ME)*, Prague, Czech Republic, 2020.

Schnittstellenimplementierung zum Aufbau einer numerischen Prozesskette in der Eisengussindustrie

Wolfgang Schlüter^{1*}, Nazar Adamchuk¹

¹Hochschule Ansbach, Residenzstraße 8, 91522 Ansbach; *wolfgang.schlueter@hs-ansbach.de

Abstract. Der Einfluss von Simulationssoftwarepaketen bei der Bauteilentwicklung nimmt zu. In der Gussindustrie werden Softwarewerkzeuge für 3-D-CAD, Material, Strukturanalyse, zerstörungsfreie Prüfung (ZfP) und Lebensdaueranalyse eingesetzt. Sie sind Teile der numerischen Prozesskette zur Entwicklung von Gussbauteilen, in der verschiedene Informationen ausgetauscht werden müssen. Derzeit fehlen aber einige Schnittstellen zwischen wichtigen Softwarepaketen. Daher wird in diesem Beitrag ein automatischer Schnittstellen-Generator als Lösung für diese Herausforderung vorgeschlagen. Der Generator benötigt nur die syntaktische Definition der Eingangsdatei und die Konvertierungsparameter in der erweiterten Backus-Naur-Form (EBNF)-Notation.

Einleitung

Numerische Simulationen als unterstützendes Werkzeug zur Entwicklung und Herstellung zeit- und kostenoptimierter Großgussbauteile erlangen in der industriellen Anwendung immer größere Bedeutung [1]. Bisher existieren Simulationstools für die Auslegung und Prüfung von Gussbauteilen meist als Insellösungen. Aktuell gibt es noch keine einheitliche Definition der Prozesskette zur Konstruktion und Auslegung von Eisengussbauteilen in der Branche. Der Informationsaustausch zwischen den einzelnen Softwaretools als Basis für eine funktionierende Prozesskette ist auf Grund von fehlenden oder nicht hinreichend dokumentierten herstellereigenen Schnittstellen nur bedingt möglich.

Hier setzt das öffentlich geförderte Projekt *DNaguss* an, in dem zwei Eisengießereien, verschiedene Softwareunternehmen und drei Forschungsinstitute zusammenarbeiten, um die einzelnen Simulationsprogramme zu einer virtuellen Prozesskette zu verknüpfen. Ein entscheidender Beitrag dazu ist die Implementierung von Schnittstellen zwischen den einzelnen Softwaretools.

1 Numerische Prozesskette

Am Anfang der Prozesskette steht die CAD-Konstruktion des Bauteils und die Auswahl der Legierung auf der

Basis einer Materialdatenbank. Da diese Daten essentiell für die Gießprozesssimulation sind, existieren die entsprechenden Importfunktionen auf Seiten der Gießprozesssimulation [2]. Die Gießprozesssimulation berechnet die Gießprozessparameter und gibt diese in einem Finit-Volumen (FV) -Methode-Netz aus. Mit einem speziellen Konvertierungstool werden die Parameter von dem FV-Netz der Gießprozesssimulation auf das Finite-Element (FE)-Netz der Strukturanalyse gemappt [3]. Die Strukturanalyse nutzt die Daten, um daraus die Verteilung der mechanischen Spannungen unter Betriebslasten zu berechnen. Das FE-Netz wird auch in der Lebensdaueranalyse benutzt, um zu bestimmen, wie lange einzelne Bauteilregionen den Betriebsbeanspruchungen standhalten können.

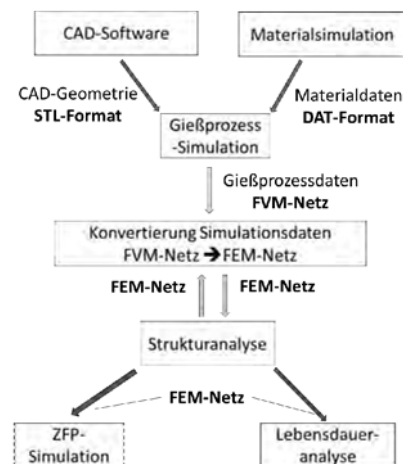


Abbildung 1: Prozesskette in der Gussindustrie

Ein ganz neues Element der Prozesskette ist die in dem Projekt *DNaguss* entwickelte Simulation zur zerstörungsfreien Prüfung (ZfP), mit der die Anwendbarkeit zerstörungsfreier Prüfmethode für ein Bauteil, auf Regionen bezogen, beurteilt werden. Die ZfP-Simulation benötigt die Geometrie des Bauteils, Materialdaten und

die orts aufgelöste Porösität des Bauteils aus den vorgelagerten Prozessschritten und liefert die regionsbezogene Anwendbarkeit von Prüfmethoden. Um weitere Daten zwischen den einzelnen Simulationstools auszutauschen und damit Rückkopplungsmechanismen in die Prozesskette einzubauen, fehlen verschiedene Schnittstellen.

2 Schnittstellen-Generierung

Für die Implementierung der Schnittstellen wurde auf der Grundlage der Erweiterten Backus-Naur Form (EBNF) ein allgemeiner Ansatz gewählt [4, 5]. Damit lässt sich ein Dateiformat metasprachlich durch Regeln der Form

```
Header = Quantity " m" "\n".
DataSet = {Temp " " Data ["\n"]}
```

mit Hilfe von Nichtterminalen wie Header, DataSet, Quantity, Temp oder Data und Terminalen wie " m" oder "\n" beschreiben. In einem ersten Schritt wurde mit dem Akzeptorgenerator ein Softwaretool entwickelt, dass auf Basis einer EBNF-Dateiformatbeschreibung die Überprüfung der syntaktischen Korrektheit der Ausgabe-datei eines in der Prozesskette vorhandenen Simulationsprogrammes prüft und gegebenenfalls einen Übertragungsfehler erkennt.

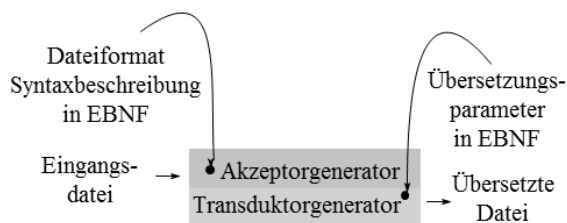


Abbildung 2: Akzeptor- bzw. Transduktorgenerator

In einem zweiten Schritt wird die EBNF-Dateiformatbeschreibung durch die Konvertierungsinformationen (hier Verweise Translate und Conv) ergänzt.

```
Header = (Quantity)(Translate) " m" "\n".
DataSet = {(Temp " " Data)(Conv) ["\n"]}
```

Über die Verweise werden kurze Codesegmente ausgeführt: Translate stellt die physikalische Größe Quantity in einer anderen Form dar, Conv bestimmt den physikalischen Wert Data zur Temperatur Temp von 20°C.

Mit dem Transduktorgenerator steht ein Softwaretool

zur Verfügung, das formatiert ausgegebene Simulationsdaten in ein anderes Format überführen kann. Benötigt wird für die Übersetzung lediglich die Grammatik-Beschreibung in EBNF-Notation, erweitert um die Konvertierungsinformationen. Dabei können auch komplexe Konvertierungen durch zusätzliche Codesegmente hinzugefügt werden. Der Vorteil des hier vorgestellten Ansatzes besteht in der einfachen Realisierbarkeit von Schnittstellen, da für die Konvertierung nur noch EBNF-Textdateien und einfachen Codesegmente benötigt werden.

3 Ausblick

Durch zusätzliche Schnittstellen soll die Prozesskette (Abb. 1) erweitert und durch Rückkopplungsmechanismen ergänzt werden. Neben der Zeit- und Kostenersparnis für den Bearbeiter ergeben sich Vorteile in der Datenintegrität und neue zusätzliche Funktionalitäten in den Softwarepaketen. Die Prozesskette in der Gussindustrie kann als Testfall für das entwickelte Konzept der Schnittstellen-Generierung dienen, welches im Erfolgsfall auch zur Erzeugung weiterer Schnittstellen zwischen Simulationsprogrammen genutzt werden kann. Im Projekt *DNAguss* wird die Prozesskette abschließend an zwei Bauteilen aus der Windenergiebranche getestet und der sich daraus ergebende Vorteil quantifiziert.

Literaturverzeichnis

- [1] Vijayaram TR, Piccardo P. Computers in Foundries. *Metallurgical Science and Technology*. 2012;30(2).
- [2] Guo Z, Saunders N, Schillé J, Miodownik A. Material properties for process simulation. *Materials Science and Engineering: A*, 2009;499(1-2):7-13..
- [3] Egner-Walter A, Dannbauer H. Integration lokaler Bauteileigenschaften gegossener Fahrwerksteile in die Betriebsfestigkeitsberechnung. 2004.
- [4] Backus JW, Bauer F, Green J, Katz C, McCarthy J, Naur P, Perlis A, Rutishauser H, Samelson K, Vauquois B, et al. Revised report on the algorithmic language ALGOL 60. *Numerische Mathematik*. 1962;4(1):420–453. 1960.
- [5] Hopcroft JE, Motwani R, Ullman JD. *Introduction to Automata Theory, Languages, and Computation (3rd Edition)*. USA: Addison-Wesley Longman Publishing Co., Inc. 2006.

Tutorial: Variantenmodellierung und automatisierte Simulationsexperimente

Thorsten Pawletta*, Hendrik Folkerts, Christina Deatcu

Forschungsgruppe Computational Eng. & Automation, Hochschule Wismar, Philipp-Müller-Str. 14,
D-23966 Wismar, Germany; *thorsten.pawletta@hs-wismar.de

Abstract. Das Tutorial führt anhand einer Fallstudie in die Variantenmodellierung und Automation von Simulationsexperimenten auf Basis eines erweiterten *System Entity Structure/Model Base* (SES/MB) Ansatzes ein. Neben der Theorie und der Softwarearchitektur werden freie SES/MB-basierte Softwarewerkzeuge und deren Integration mit OpenModelica und MATLAB/Simulink vorgestellt.

Einleitung

Technische Systeme, die heute unter Schlagworten wie Internet der Dinge, Cyber-Physical-Systems oder Industrie 4.0 zusammengefasst werden, sind durch eine hohe Komplexität und Variabilität gekennzeichnet. Im Software-Engineering (SE) definieren Capilla et al. [1] Variabilität als die Fähigkeit, ein System je nach Einsatzzweck und Zielstellung zu konfigurieren, zu erweitern oder anzupassen. Demgemäß ist mit Variabilität zumeist eine hohe Variantenvielfalt verbunden. Zur Beherrschung vielfältiger Softwarevarianten wurde im SE, in Analogie zu Produktlinien in der Fertigung, das Konzept der *Software Product Lines* (SPLs) entwickelt. Das SPL-Engineering umfasst Methoden und Werkzeuge zur Erstellung unterschiedlicher Softwarevarianten auf Basis einer gemeinsamen Plattform.

Wie im SE stellt das Management von Variabilität und Variantenvielfalt eine Herausforderung im Bereich der Modellbildung und Simulation (M&S) dar. Das Pendant zu SPLs im SE sind Modellfamilien in der M&S. Nach Zeigler [2] ist eine Modellfamilie, die Menge aller Modelle der Entwurfsalternativen, die sich unter Beachtung der gewählten Systemgrenzen und der verfolgten Entwurfsziele ergeben. Die konkrete Herausforderung besteht demgemäß: (i) in der Modellierung einer Modellfamilie, (ii) der Auswahl eines oder mehrerer Kandidatenmodelle, (iii) der Generierung eines oder mehrerer ausführbarer Simulationsmodelle und (iv) der Bewertung der Simulationsergebnisse. Das Tutorial zeigt wie, aufbauend auf dem von Zeigler [3] eingeführten und durch

verschiedene Kollegen kontinuierlich weiterentwickelten Ansatz der *System Entity Structure / Model Base* (SES/MB) ([4], [5]), die gestellten Anforderungen gemeistert werden können. Darüber hinaus wird gezeigt, wie simulatorunabhängige MBs aufgebaut werden können und wie auf Basis einer erweiterten SES/MB-Architektur automatisierte Experimente mit Modellfamilien durchführbar sind.

Im Tutorial steht nicht die Vollständigkeit der einzelnen SES/MB-Methoden mit allen Erweiterungen im Vordergrund. Es geht primär um das grundlegende konzeptionelle Verständnis und die prinzipiellen Möglichkeiten zur Automation von Simulationsexperimenten mit Modellfamilien, einschließlich der Verwendung unterschiedlicher Simulatoren.

Aufbau des Tutorials

Das Tutorial führt schrittweise anhand eines Fallbeispiels in die Variantenmodellierung und Automation von Simulationsexperimenten auf Basis eines erweiterten SES/MB-Ansatzes ein ([6], [7]). Nebenläufig zur Theorie wird das Fallbeispiel schrittweise implementiert. Dabei wird auf freie SES/MB-basierte Softwarewerkzeuge eingegangen und deren Integration mit Simulationsumgebungen am Beispiel von OpenModelica und MATLAB/Simulink gezeigt. Die freien Werkzeuge stehen unter GitHub zur Verfügung ([8], [9]). Das Tutorial ist in sieben Schwerpunkte unterteilt und endet mit einem Fazit.

Fallbeispiel. Es werden zwei mögliche Strukturvarianten eines Regelsystems eingeführt. Es gilt, für unterschiedliche Regelziele und Nebenbedingungen die möglichst minimale Struktur mit dazugehöriger Parametrierung per Systemsimulation zu finden.

SES/MB-basierte Modellierung. Es werden die grundlegenden Prinzipien der SES/MB-basierten

Modellierung von Modellfamilien eingeführt. Danach werden am Fallbeispiel die simulatorunabhängige Spezifikation von Modellvarianten, Modellstrukturen und Modellparametrierungen mit einer SES diskutiert. Anschließend wird für das Fallbeispiel eine simulatorabhängige MB unter Verwendung von dynamischen Modelica Basiskomponenten aufgebaut.

Implementierung einer SES. Ausgehend von der für das Fallbeispiel spezifizierten SES werden ein Python-basierter SES-Editor und dessen Integration mit einem separaten SES-Viewer gezeigt. Dabei werden fortgeschrittene SES-Methoden praktisch demonstriert, wie zum Beispiel das Mergen von SES-Modellen. Für den Austausch mit anderen Werkzeugen werden SES-Modelle im JSON- oder XML-Dateiformat gespeichert.

Modellauswahl und -generierung. Es wird die Ableitung von konkreten Modellkonfigurationen aus einer SES und die Generierung von simulatorspezifischen Modellen gezeigt. Es folgt eine softwaretechnische Demonstration für das Fallbeispiel, wobei ausführbare Modelle für OpenModelica generiert werden.

Die freie MATLAB SES Toolbox. Aufgrund der weiten Verbreitung von MATLAB/Simulink wurden spezielle SES/MB-basierte Werkzeuge entwickelt, welche als APP in das System MATLAB/Simulink integrierbar sind. Am Fallbeispiel werden unter Verwendung der speziellen Tools alle Schritte von der Modellierung der simulatorunabhängigen SES sowie dem Aufbau einer MB mit Simulink bis zur Generierung und Ausführung von Simulink-Modellen aufgezeigt.

Organisation simulatorunabhängiger MBs. Basierend auf dem Functional-Mock-up-Interface (FMI) wird ein Ansatz zum Aufbau simulatorunabhängiger MBs aufgezeigt und am Fallbeispiel softwaretechnisch demonstriert.

Automation von Simulationsexperimenten. Es wird eine erweiterte SES/MB-Architektur mit Komponenten zur vollständigen Automation von Simulationsexperimenten vorgestellt. Ausgehend von einer Experimentspezifikation mit den Entwurfszielen und geltenden Nebenbedingungen werden: (i) zulässige Modellkonfigurationen aus der SES abgeleitet, (ii) simulatorspe-

zifische oder FMI-basierte Modelle generiert, (iii) mit einem Simulator ausgeführt und (iv) die Simulationsergebnisse analysiert. In Abhängigkeit der erzielten Ergebnisse werden weitere Modellkonfigurationen studiert oder das Experiment beendet. Anhand des Fallbeispiels wird die Automation softwaretechnisch demonstriert.

Fazit. Abschließend werden bestehende Defizite und mögliche Weiterentwicklungen des SES/MB-Ansatzes sowie der Software-Tools im Kontext der aktuellen Forschung diskutiert.

Danksagung

Wir danken Bernie Zeigler und Jerzy Rozenblit für die wertvollen Diskussionen und die Förderung unserer Forschung.

References

- [1] Capilla R, Bosch J, Kang K-C, editors. *Systems and Software Variability Management: Concepts Tools and Experiences*. Heidelberg, New York et al.: Springer, 2013.
- [2] Zeigler B.P, Praehofer H, Kim T.G. *Theory of Modeling and Simulation*. 2nd Ed. San Diego: Academic Press; 2000.
- [3] Zeigler B.P. *Multifaceted Modeling and Discrete Event Simulation*. 1st Ed. London: Academic Press; 1984.
- [4] Rozenblit J.W, Zeigler B.P. Representing and constructing system specifications using the system entity structure concepts. In *Proc. of Winter Simulation Conference*; 1993; Los Angeles, CA, USA, 604-611.
- [5] Zeigler B.P., Sarjoughian H.S. *Guide to Modeling and Simulation of Systems of Systems*. London: Springer Pub.; 2013
- [6] Pawletta T, Durak U, Schmidt A. Modeling and Simulation of Versatile Technical Systems Using an Extended System Entity Structure / Model Base Infrastructure. In: Zhang L, Zeigler B.P, Laili Y editors. *Model Engineering for Simulation*. 1st Ed. London: Academic Press / Elsevier; 2019. 393-418.
- [7] Folkerts H, Pawletta T, Deatcu C. Model Generation for Multiple Simulators Using SES/MB and FMI. In *25th ASIM Symposium Simulationstechnik (SST)*; 2020 Okt; virtuell. ARGESIM Report 59. 13-20, doi: 10.11128/arep.59.a59003
- [8] Folkerts H. *Python-based SES/MB Toolset*. <https://github.com/hendrikfolkerts>
- [9] CEA Wismar. *SES Toolbox for MATLAB*. https://github.com/cea-wismar/SES_Tbx_Matlab

Real-Time Detection and Classification for a 360°-Camera Using a YOLO Algorithm

Tetiana Lavrenko^{1*}, Ayman Ahmed¹, Vladimir Prokopenko¹, Thomas Walter¹, Hubert Mantz¹

¹Institute for Mechatronic and Medical Engineering, Ulm University of Applied Sciences, Albert-Einstein-Allee 55, 89081 Ulm, Germany; **tetiana.lavrenko@thu.de*

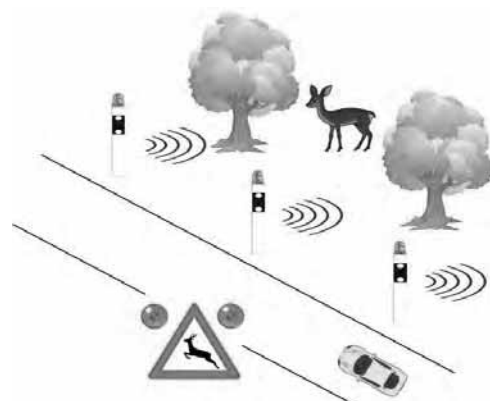
Abstract. The focus of this paper is the detection and classification of different objects in real time with the help of a 360°-camera. YOLO, a computer vision algorithm, is to be used to perform both the localization and classification of the objects present in the equirectangular panoramic images. The algorithm will be extended in such a way that the angles and directions with respect to the camera are assigned to the detected objects. The results of this work can contribute to enhanced road safety at the locations where many traffic accidents take place due to suddenly appearing road users.

Introduction

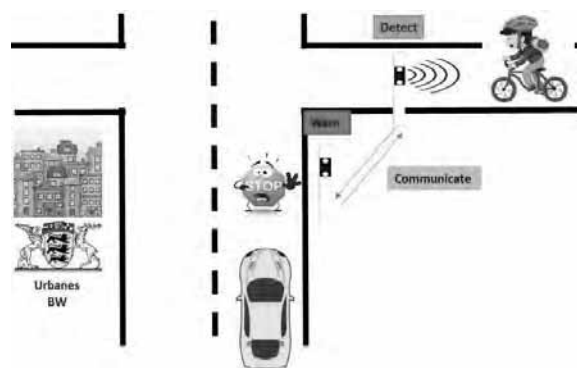
A high level of mobility is a basis of highly developed society and economy. However, the current development of the road accidents statistics points out at the fact that vulnerable road users will have to remain a focus of a road safety work in the coming years. [1] As reported by the Federal Statistical Office [2], one in seven people killed in road traffic in 2019 was on a bicycle. If car drivers were involved in a cycling accident with personal injury, bicyclists were mainly to blame in only 23.4% of the cases. Furthermore, traffic accidents on busy roads caused by wild animals are becoming increasingly common. Every year, the Federal Statistical Office reports more than 270,000 accidents involving wild animals. According to the German Hunting Association (DJV), more than one million wild animals are killed in accidents yearly. Most collisions involve deers. [3] Thus, additional measures have to be taken to enhance safety on the roads. One of the approaches would be to inform drivers in advance about a potentially dangerous situation at the spots with increased accident risk where available warning systems fail to inform early enough about suddenly appearing animals, cyclists or pedestrians on the road.

From the aforementioned considerations, two case

scenarios can be distinguished: wild animal crossings in the forestry regions and the traffic intersections with obscured side views as has been depicted in Figure 1.



(a) wild crossing



(b) intersection assistant

Figure 1: Scenarios of interest for the discussed application.

Modern computer vision algorithms in combination with 360°-cameras can contribute to the development of new road safety concepts. Nowadays 360°-cameras are experiencing enormous rise related to the increasing pop-

ularity of virtual reality products. They produce 3D-images of the surroundings, which can be stored and/or converted into 2D-projections. These new data formats of videos and images imply new challenges as well as provide numerous possibilities for computer vision and image processing. All-around monitoring systems based on spherical or panoramic recordings could become a new basis for smart road infrastructure covering both case scenarios mentioned above.

In this paper, an experimental setup for real-time detection and localization for a 360°-camera using a YOLO algorithm will be presented. In the context of improved road safety, the computer vision algorithm will be extended with the possibility to estimate an angle of arrival of detected objects assigning them additionally relative geographical directions with respect to the camera's position.

1 Experimental

The discussed setup consists of a GARMIN Virb 360 camera, an Nvidia board for GPU-enhanced artificial intelligence (AI) projects and a YOLO-algorithm based on [4].

The Virb 360 camera is able to record all-around images and videos, which can be also viewed with virtual reality glasses for the “in the middle of it” experience [5]. It has two optical sensors, front and rear, with resolution of 12 Mpx and the objective with the extra wide opening angle of 200°. The maximum resolution of the camera is 5.7k, but for real-time recording and streaming, it is downgraded to 4k.

The Nvidia Jetson Nano developer kit is a small and powerful computer board able to run multiple neural networks in parallel such as image classification, object recognition, segmentation, etc [5]. The Jetson Nano delivers 472 GFLOPs to enable modern AI algorithms to run quickly. This computational capacity can also process data from numerous high-resolution sensors simultaneously with full analysis capabilities. The high performance is facilitated by GPUs – Nvidia graphics processor with the Maxwell architecture and 128 Nvidia CUDA computing units.

Deciding upon which computer vision algorithm to choose for the specified task, the following aspects have been considered. Self-explanatory that a corresponding detection speed of a chosen algorithm has to satisfy real-time requirements as a street situation may change rap-

idly. Moreover, a multi-object detection capability represents another challenge by itself as well as concerning both a speed and accuracy of the detection. Furthermore, understanding a complete image is of importance, as this should increase the overall accuracy of the object classification and detection.

YOLO is a computer vision algorithm able to detect and localize objects in real time. It is based on the single neural network called Darknet described in [6]. The advantages of this algorithm are the following: it is fast and easy to set up; open source, therefore available for everyone; can be used with other frameworks and libraries such as OpenCV, TensorFlow, PyTorch, etc.; and is highly accurate.

In the discussed experiments, YOLOv3 has been used. The comparison of the algorithm to other detectors has been clearly demonstrated with performance metrics in [6]. The feature extraction network of YOLOv3 employs a hybrid approach between the network used in YOLOv2 [7] and the one of the residual network family ResNet [8]. The phenomenon of residual networks lies in short circuiting shallow to deep layers, also known as shortcuts, resulting in deep neural networks without performance degradation problem. The main conclusion of testing in [6] is that the classifier of YOLOv3 demonstrates the highest measured floating-point operations per second (1457 vs. 1090 billion FLOP/s, the best result of the ResNet family). On the application level, it means that the network uses the GPU resources more optimally compared to the ResNet, thereby increasing own efficiency and speed. Intersection over Union (IOU) is a parameter, which describes the precision of the predicted bounding boxes of the detector network. The detection metrics of mean average precision (mAP) at IOU of 0.5 describing the accuracy vs. speed tradeoff confirms that YOLOv3 also has significant benefits compared to other detection systems demonstrating shortest inference time during testing [6].

The YOLOv3 classification network is trained on the COCO dataset (Common Objects in COntext) that covers 80 different object categories. Among them are cars, trucks, pedestrians, bicyclists that make it possible to implement this detection algorithm in the urban environment as it is.

2 Results

As has been mentioned above, the presented setup aims

at improving road safety in such way that different vulnerable road users, such as pedestrians, bicyclists, wild animals, etc., are detected and localized in the space by an additionally calculated angle/direction of arrival with respect to the camera location. This information can be

analyzed together with the location of an approaching car, truck or any heavy vehicle with a further decision to send or not a warning signal to the drivers.



Figure 2: Transfer learning using a YOLO algorithm for detecting deers.

2.1 Animal Detection with YOLO Using Transfer Learning

Considering a scenario with a hotspot of wild animal crossings depicted in Figure 1a, the YOLO algorithm can be retrained for required animal classes in case if the algorithm has not been trained for them initially. It is the case for detecting a deer, as this animal category is not present in the COCO dataset. Using pre-trained networks for further applications saves training time and requires lesser amount of training data. This approach is widely used and well known as transfer learning.

In this work, Darknet was retrained using the Google Colab. It provides the user with sufficiently powerful hardware to perform complex calculations. The training execution is done with commands from the C and Python programming languages. The neural network was trained on 3500 images of deers. The images come from the database "Open Images Dataset V6" under the link

<https://storage.googleapis.com/openimages/web/download.html>. The achieved overall accuracy of detection for the category deer was 97.77%.

The result of transfer learning applied to the YOLO algorithm can be seen in Figure 2. The deer have been detected with high confidence scores and successfully tracked during the whole measurement time. The possibility of detecting multiple objects in one frame is an important advantage of the algorithm. The presented video has been recorded using a normal camera (GoPro Hero 8 [9]), therefore the angle assignment has not been implemented in this example.

2.2 Angle Assignment

Using a 360°-camera provides numerous possibilities for further image and data analysis. Transforming a spherical image into a cylindrical or equirectangular projection re-

tains information about the surroundings in every viewing direction. Such projection ultimately can be treated as a normal image, which has a particular relationship between an image pixel and a spatial direction. Figure 3

schematically describes the approach to estimate the angle of arrival of an object (in the discussed figure a person) with respect to the camera position. The reference point (0° north) is assumed to be in the middle of an



Figure 3: Example of the object detection extended with an angle of arrival.

image (reading 0.5), which is normalized to one (1) in both vertical (height) and horizontal (width) directions. The coordinates of the bounding box predicted by the YOLO algorithm are used afterwards to estimate the object position at every time point resulting in a real-time object tracking. The spatial location of the detected object is equal to the midpoint of the bounding box calculated from its upper left (x_1, y_1) and bottom right (x_2, y_2) coordinates. The approach can be validated by overlaying an image with the readings of the digital compass in-built in the camera as can be seen in the bottom part of Figure 3.

2.3 Intersection Assistant

Figure 4 demonstrates the measurement results in the urban surroundings corresponding to the scenario in Figure 1b. The camera was located at the corner of a pedestrian

path and a street. The algorithm could successfully detect and localize two pedestrians and two cars that were moving at different angles to each other. The full detection information has been saved into a .txt-file. This file can be used for further data analysis as it contains a current video frame number, a detected category, an angle of arrival as well as an exact timestamp of the detection. At a closer look, one can see that a female pedestrian and a dark car had perpendicular movement trajectories. At an intersection where the driver's view could be obscured due to a construction site or some greenery, this could lead to a potentially dangerous situation. From these considerations, a warning system consisting from a 360° -camera to monitor the surroundings and a computer vision algorithm able to analysis a situation in a global context could contribute to an improved safety at the roads both in the urban environment as well as in the forestry areas.



Figure 4: Object detection using the YOLO algorithm extended with an angle of arrival.

3 Discussion and Conclusion

Safe and sustainable mobility in forested areas as well as in urban surroundings can be ensured by developing smart road infrastructure to reduce the number of car accidents in connection with vulnerable road users. The ideal system would be a setup that only warns drivers when there is a high risk of a potentially dangerous situation to take place. As could be seen from the results presented in this paper a combination of a 360°-camera and a YOLO algorithm can solve this task.

The advantage of the system is its ability to detect and classify multiple objects in real time. By analyzing consecutive frames, a movement direction of an object towards or away from the camera can be estimated. However, the disadvantage is a lack of information on the distance between detected objects and the camera. The respective distances can be provided by integrating a radar sensor in the system, which depending on its modulation type can measure the distance, velocity and direction of objects in front of it. The integration of a radar sensor is planned as a next step in this project. A further advantage of having a radar sensor is that this system should work reliably irrespective of environmental conditions, that is,

at night and in the presence of fog when an optical camera is of no use. Moreover, the detection results of two sensors can be compared, and thereby the accuracy of a warning signal can be enhanced. However, the challenge is that the radar measurement data is difficult to interpret in order to classify different road users. Animal walking, similarly to a human movement, is a complex process that comprises different movement patterns of single body parts. This implies a complex analysis of single movement components resulting in the unique radar features that distinguish one road user from another. Therefore, by combining an optical camera and a radar sensor, the difficulty of interpreting the radar measurements can be overcome. The camera recordings can serve as the ground truth for the further analysis of the unique radar features which can be used afterwards as an input for a classification algorithm using a supervised learning. The measurement results of the radar-based detection system will be published elsewhere.

The application for such a real-time detection system combining both optical and radar sensors can be wide-ranging in the context of a smart and sustainable city. One can start from a basic idea to collect mobility data in public urban surroundings and make it available on a data platform for agile urban planning. From another side, the system can be used more specifically for traffic counting

of particular road users irrespective it is urban surroundings or forestry rural regions. Moreover, the installation and control of adaptive lightning along the streets can also be supported by the real-time detection system. Light duration can be adjusted automatically depending whether a fast moving bicycle or a strolling pedestrian has been detected passing by. Within the framework of the InnoSüd project the bicycle counting system in the city Ulm is to be installed based on a real-time detection system combining both optical and radar sensors.

References

- [1] adac.de, 10.12.2020. [Online]. Available: <https://www.adac.de/news/bilanz-verkehrstote/#:~:text=Der%20bisherige%20Tiefststand%20lag%20nach,Minus%20von%2013%2C2%20Prozent..> [Visited on 11. 02. 2021].
- [2] destatis.de, 19. 08. 2020. [Online]. Available: https://www.destatis.de/DE/Presse/Pressemitteilungen/2020/08/PD20_N049_46241.html. [Visited on 11. 02. 2021].
- [3] adac.de, 23. 04. 2020. [Online]. Available: <https://www.adac.de/verkehr/verkehrssicherheit/tiere/wildunfaelle/#:~:text=Jedes%20Jahr%20meldet%20das%20Statistische,Wildtiere%20bei%20Unf%C3%A4llen%20ums%20Leben..> [Visited on 11. 02. 2021].
- [4] AlexeyAB, github.com, [Online]. Available: <https://github.com/AlexeyAB/darknet>. [Visited on 11. 02. 2021].
- [5] Nvidia, [Online]. Available: <https://developer.nvidia.com/embedded/jetson-nano-developer-kit>. [Visited on 11. 02. 2021].
- [6] A. F. Joseph Redmon, „YOLOv3: An Incremental Improvement,“ arXiv:1804.02767, 2018.
- [7] J. Redmon und A. Farhadi, „YOLO9000: Better, Faster, Stronger,“ arXiv, 2016.
- [8] K. He, X. Zhang, R. Shaoqing und J. Sun, „Deep Residual Learning for Image Recognition,“ arXiv, 2015.
- [9] gopro.com, [Online]. Available: <https://gopro.com/de/lu/shop/hero8-black/tech-specs?pid=CHDHX-801-master>. [Visited on 12. 02. 2021].
- [10] VIRB® 360 Owner’s Manual,“ [Online]. Available: http://static.garmin.com/pumac/Virb_360_OM-EN.pdf. [Visited on 11. 02. 2021].

Cloudbasierte V2X-Kommunikation zur dynamischen Zielführung im vernetzten Verkehr

Sören Scherler^{1*}, Lars Rüdiger Kelm², Sven Jacobitz¹, Florian Pramme², Gert Bikker², Xiaobo Liu-Henke¹

¹Ostfalia Hochschule für angewandte Wissenschaften, Fakultät Maschinenbau, Institut für Mechatronik, Salzdhallumer Str. 46/48, 38302 Wolfenbüttel, Deutschland; *so.scherler@ostfalia.de

²Ostfalia Hochschule für angewandte Wissenschaften, Fakultät Informatik, Institut für Verteilte Systeme, Am Exer 2, 38302 Wolfenbüttel, Deutschland

Abstract. In diesem Beitrag werden Entwurf und Verifikation einer Netzwerkinfrastruktur zur cloudbasierten V2X-Kommunikation und dynamischen Zielführung im vernetzten Verkehr dargestellt. Mithilfe der entworfenen Infrastruktur wird die beschränkte Reichweite der WLAN-basierten V2X-Kommunikation nach IEEE 802.11p erhöht, sodass dynamische Informationen über Störungen des Verkehrsflusses in einem weiten Umkreis vorliegen. Eine Schnittstelle auf Fahrzeugebene zur Integration dieser Informationen in das Kartenmaterial ermöglicht eine dynamische Zielführung, welche frühzeitig optimale Alternativrouten berechnet.

Einleitung

Motivation

Im Teilprojekt *Intelligente Elektrofahrzeuge mit Range Extender in Verkehrssystemen mit Fahrzeug 4.0 (iREX 4.0)* des vom Nds. Ministeriums für Wissenschaft und Kultur sowie der VolkswagenStiftung geförderten Verbundprojekts *Zukünftige Fahrzeugtechnologien im Open Region Lab (ZuFOR)* werden automatisierte Elektrofahrzeuge mit Brennstoffzellen-Range-Extender untersucht. Das iREX 4.0 zielt auf den zeit- und energieoptimierten, prädiktiven Fahrbetrieb im vernetzten Verkehr [1].

Um dieses Ziel zu erreichen, werden Optimierungspotentiale in der Zielführung („Optimale Route zur Erfüllung der Fahraufgabe“), der Bahnplanung („Optimale Fahrzeugführung auf der gewählten Route zur Erfüllung der Fahraufgabe“) und dem Energiemanagement („Optimale Verteilung der zur Erfüllung der Fahraufgabe notwendigen Leistung auf Batterie und Brennstoffzelle“) fokussiert. Im vorliegenden Beitrag wird mit der Zielführung das erste Optimierungspotential adressiert.

Problem- und Zielstellung

Die dynamische Zielführung berücksichtigt zum Finden optimaler Routen neben statischen Karten zunehmend dynamische Echtzeit-Verkehrsinformationen, die über etablierte Kanäle wie dem Traffic Message Channel (TMC) oder dem Nachfolgeprotokoll der Transport Protocol Experts Group (TPEG) bereitgestellt werden. Zusätzlich bietet die WLAN-basierte V2X-Kommunikation nach IEEE 802.11p in ad-hoc-Netzwerken weitere Informationen zur Verbesserung der Zielführung.

Die Problemstellung liegt in der auf einige 100 m beschränkten Reichweite der WLAN-basierten V2X-Kommunikation, sodass relevante Informationen erst kurzfristig und zu spät für eine vorausschauende Alternativroutenplanung berücksichtigt werden können. Die Zielstellung der in diesem Beitrag dargestellten Forschungstätigkeiten liegt in der Reichweitenerhöhung der V2X-Kommunikation durch Entwicklung einer Netzwerk- und Cloudinfrastruktur und deren Integration in die dynamische Zielführung eines Fahrzeugs.

Struktur des Beitrags

Als Grundlage dieses Beitrags wird in Abschnitt 1 ein Überblick über den Stand des Wissens zu vernetzten Verkehrssystemen, zu Technologien der V2X-Kommunikation sowie zur dynamischen Zielführung dargestellt. Die eingesetzte ganzheitliche, mechatronische Entwicklungs- und Absicherungsmethodik wird in Abschnitt 2 vorgestellt. In Abschnitt 3 erfolgt die Konzeption einer Netzwerkinfrastruktur zur cloudbasierten V2X-Kommunikation und dynamischen Zielführung. Der Entwurf der Netzwerk- und Cloudinfrastruktur wird in Abschnitt 4 und die Erweiterung der dynamischen Zielführung um die cloudbasierte V2X-Kommunikation in Abschnitt 5 dargestellt. Abschnitt 6 beschreibt die Verifikation und

Validierung anhand einer Pilotanwendung und Abschnitt 7 ein Resümee sowie einen Ausblick.

1 Stand des Wissens

In diesem Abschnitt wird ein Überblick über den Stand des Wissens dargestellt.

1.1 Vernetzte Verkehrssysteme

Vernetzte Verkehrssysteme basieren auf den u.a. in der EU-Richtlinie 2010/40/EU definierten Intelligent Transportation Systems (ITS). Sie beinhalten Fahrzeuge, Infrastrukturelemente bzw. Road Side Units (RSUs) und weitere (nichtmotorisierte) Verkehrsteilnehmer, welche drahtlos miteinander vernetzt sind und Telemetriedaten austauschen. Diese können zur Verbesserung des eigenen Betriebs und auch für kooperative Funktionen (z.B. für Kreuzungsmanagementsysteme, vgl. [2]) verwendet werden [3]. Die drei wesentlichen Ziele eines ITS sind nach [4] die Steigerung von Verkehrseffizienz, Sicherheit und Energieeffizienz. Essenzielle Grundlage zur Zielerfüllung ist die Vernetzung mittels drahtloser V2X-Kommunikation, die durch die im folgenden Abschnitt 1.2 vorgestellten Technologien realisiert werden kann.

1.2 Technologien zur V2X-Kommunikation

Die Realisierung der V2X-Kommunikation erfolgt entweder mit der Dedicated Short Range Communication (DSRC), einer WLAN-basierten Kommunikation nach IEEE 802.11p in flexiblen ad-hoc-Netzwerken, oder mit der Cellular-V2X-Communication (C-V2X), einer auf LTE oder 5G basierenden Mobilfunktechnologie.

Die DSRC benötigt im Gegensatz zu C-V2X keine eigene Infrastruktur, wodurch die Leistungsfähigkeit unabhängig von Netzabdeckung und -auslastung ist [5], keine Investitionskosten anfallen und keine Abhängigkeit vom flächendeckenden Ausbau des 5G-Netzes besteht. Nach [6] beträgt die maximale, effektive Reichweite der DSRC unter Sichtkontakt ca. 1 km und unter Nicht-Sichtkontakt, wie sie im urbanen Umfeld i.d.R. vorliegen, ca. 300 m. C-V2X hingegen verfügt nach [7] durch Nutzung des Mobilfunknetzes über eine unbegrenzte Reichweite und auch eine höhere Bandbreite bei Kommunikation über mehr als 300 m, sodass mehr Daten über eine größere Distanz ausgetauscht werden können.

Für die V2X-Kommunikation, insb. für die DSRC, existiert eine Vielzahl genormter Nachrichtentypen [8]. Das European Telecommunications Standards Institute

(ETSI) spezifiziert für den europäischen Raum bspw. die periodisch versendete Cooperative Awareness Message (CAM, [9]) und die ereignisbasiert versendete Decentralised Environmental Notification Message (DENM, [10]). CAM beinhalten Informationen über Fahrzeugart und aktuelle Fahrzeugzustände (z.B. Geschwindigkeit, Position), um z.B. Kollisionen zu vermeiden. DENM enthalten Informationen über Ereignisse, welche Verkehrssicherheit oder -zustand beeinträchtigen.

Potentiale zur Reichweitenerhöhung liegen im Multi-Access-Edge-Computing (MEC), welches Netzwerkstrukturen mit zwischen einer zentralen Cloud sowie mehreren Netzknoten (Edges, im Kontext dieses Beitrags RSUs) aufgeteilter Rechenleistung vorsieht [11]. Vorteilhaft ist einerseits, dass die Verarbeitung von V2X-Nachrichten dezentral und lokal in der RSU erfolgen kann, und andererseits, dass die Netzwerkanbindung der RSUs an die Cloud eine Verteilung der V2X-Nachrichten über große Distanzen ermöglicht. [12] stellt vielfältige Anwendungsfälle des MEC für die V2X-Kommunikation dar.

1.3 Dynamische Zielführung

Algorithmen zur Zielführung suchen in gerichteten und gewichteten Graphen, welche ein Straßennetzwerk mathematisch beschreiben, einen zeit-, distanz- oder energieoptimalen Pfad zwischen einem Start- und einem Zielknoten. In vielen Anwendungen werden der Dijkstra-Algorithmus [13] oder eine der zahlreichen Weiterentwicklungen zur Effizienzsteigerung (vgl. [14]) eingesetzt. Aktuelle Forschungstätigkeiten betrachten zudem stochastische Verfahren wie die von [15] vorgestellte Ant Colony Optimization oder die von [16] dargestellten Ansätze der Künstlichen Intelligenz mit maschinellem Lernen.

Die Güte der besten Route hängt bei allen Ansätzen von der Aktualität und Vollständigkeit des zugrundeliegenden Kartenmaterials ab. Der Informationsgehalt einer Karte wird anhand des in Abbildung 1 dargestellten Modells der Local Dynamic Map (LDM) in vier Schichten in Abhängigkeit ihrer Dynamik klassifiziert.

Die beiden unteren Schichten (Type 1/2) beinhalten statische Informationen, die i.W. das Straßennetz und die Verkehrsinfrastruktur beschreiben.

Die Type3-Schicht, Grundlage der dynamischen Zielführung, enthält transient-dynamische Informationen über den Straßenzustand (Glätte, Nässe, etc.), Schaltphasen von Lichtsignalanlagen sowie das Verkehrs- und Unfallgeschehen. Diese Informationen werden durch die

etablierten Technologien des TMC und seines Nachfolgeprotokolls TPEG bereitgestellt. Der TMC basiert auf analogem Rundfunk und weist eine geringe Datenrate von ca. 10 Meldungen pro Minute auf. Zudem kann für eine Meldung nur bestimmt werden, zwischen welchen zwei Anschlussstellen von Autobahnen oder Bundesstraßen sie liegt. Per TPEG werden Meldungen digital per Internet, Mobilfunk oder drahtlosen Netzwerken verteilt. Mit digitalem Radio (DAB) lassen sich bspw. mehr als 3000 Meldungen pro Minute mit eindeutig definierter Position in WGS84-Koordination versenden. [17]

Die vierte Ebene (Type 4) enthält hochdynamische Informationen mit einem Horizont weniger Sekunden wie Fahrzeuggeschwindigkeiten und -positionen oder Fahrzeugzustände (z.B. Notbremsung oder ABS-Eingriff). Diese Informationen werden per V2X-Kommunikation von anderen Fahrzeugen oder der Infrastruktur in einem lokal begrenzten Umkreis bereitgestellt oder durch die eigene Umfeldsensorik ermittelt.

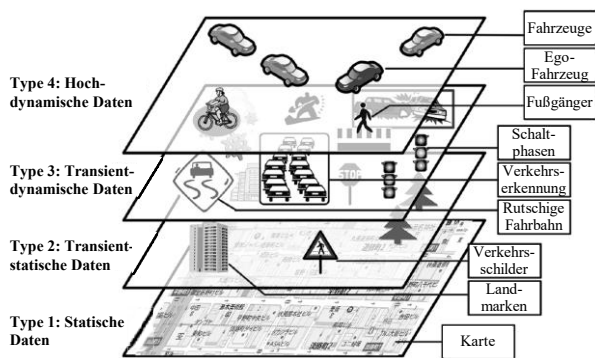


Abbildung 1: Schichtenmodell einer LDM nach [18].

1.4 Fazit

Die Vernetzung in Verkehrssystemen bietet neue Informationsquellen, welche zur Optimierung des Fahrbetriebs verwendet werden können. Die V2X-Kommunikation scheint eine sinnvolle Ergänzung der LDM auf Type3-Schicht zur Optimierung der dynamischen Zielführung zu sein. Dies gilt insbesondere für kurzfristig aufgetretene Ereignisse, welche noch nicht durch TMC oder TPEG erfasst wurden. Aufgrund des in Deutschland aktuell noch laufenden Ausbaus des 5G-Netzwerks und der nicht flächendeckenden Verfügbarkeit wird die DSRC als Kommunikationstechnologie in diesem Beitrag zugrunde gelegt. Problem für die dynamische Zielführung ist die begrenzte Reichweite der DSRC, allerdings werden MEC-Ansätze als aussichtsreich zur signifikanten Reichweitenerhöhung betrachtet.

2 Methodik

2.1 Mechatronische Entwicklungsmethodik

Die durchgängig modellbasierte und verifikationsorientierte Funktionsauslegung und -absicherung vernetzter mechatronischer Systeme (vgl. Abbildung 2) nach [19] hat sich in zahlreichen Anwendungen in Forschung und Industrie als zeit- und kosteneffizient erwiesen.

Ausgehend von Anforderungen und Lastenheft erfolgt die Model-in-the-Loop-Simulation (MiL), in der Funktionen ohne Anspruch auf Echtzeitfähigkeit entwickelt und an einem Systemmodell erprobt werden. Aus den simulativ erprobten Funktionen wird im Rahmen der Software-in-the-Loop-Simulation (SiL) mittels automatischer Code-Generierung ein ausführbarer Target-Code erzeugt, welcher wiederum mit einem Systemmodell auf Funktion und Fehler getestet wird. Daraufhin erfolgt die Hardware-in-the-Loop-Simulation (HiL), bei welcher ein um physische Teilkomponenten ergänztes, echtzeitfähiges Systemmodell zur Funktionsabsicherung und -optimierung verwendet wird. Begleitet wird dieses Vorgehen durch stetige Messungen an realen Systemen, um fortlaufende Verifikationen und Validierungen der Modelle und der ausgelegten Regelalgorithmen durchzuführen.

Mithilfe dieser Methodik wird eine durchgängige Entwicklung basierend auf virtuellen Prototypen bis zum fertigen Produkt sichergestellt. Dieser Beitrag stellt aus der Methodik schwerpunktmäßig die Verifikation und Validierung der cloudbasierten V2X-Kommunikation zur dynamischen Zielführung unter Echtzeitbedingungen mittels HiL-Simulation dar (vgl. Abschnitt 6).

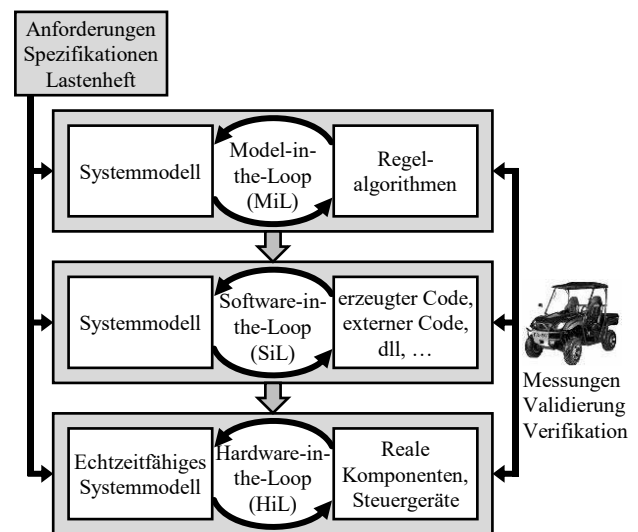


Abbildung 2: Durchgängig modellbasierte, verifikationsorientierte Entwicklungsmethodik nach [19].

2.2 Relevante Prüfinfrastruktur

Zur Durchführung von HiL-Simulationen unter Echtzeitbedingungen werden ein mobiler HiL-Prüfstand (Simulation des Ego-Fahrzeugs) sowie eine waveBEE®plus der Fa. Nordsys (Verkehrssimulation) verwendet.

Der mobile HiL-Prüfstand (vgl. Abbildung 3) verfügt über ein Echtzeitsystem der Fa. dSPACE (MicroLabBox - MLBX), auf dem Echtzeitanwendungen (z.B. die dynamische Zielführung) ausgeführt werden. Die MLBX ermöglicht zudem die Anbindung externer IO sowie verschiedener Bussysteme. Weiterhin verfügt er über eine V2X-Einheit zur DSRC mittels WLAN (IEEE 802.11p), in der eine integrierte GPS-Antenne zur Positionsbestimmung enthalten ist. Ein Nachrichtenkonverter bildet die Schnittstelle zum Echtzeitsystem. Er kodiert die von der Echtzeitanwendung generierten V2X-Nachrichten für den Versand und dekodiert die empfangenen Nachrichten, sodass Ego-Position und V2X-Daten durch die Echtzeitanwendung genutzt werden können.

Die Echtzeitanwendung wird in Matlab/Simulink auf dem Prüfstands-PC programmiert und im Anschluss mittels automatisierter Codegenerierung auf die MLBX übertragen. Der Prüfstands-PC dient zudem als Human-Machine-Interface (HMI), durch welches der Nutzer den V2X-Prüfstand überwachen und steuern kann.

Die waveBEE®plus ist ein V2X-Umfeldgenerator für DSRC-Nachrichten. Dieses wird zum Test von auf DSRC basierenden Funktionen genutzt, indem virtuelle Verkehrsszenarien definiert und reale V2X-Nachrichten der virtuellen Verkehrsteilnehmer versendet werden.

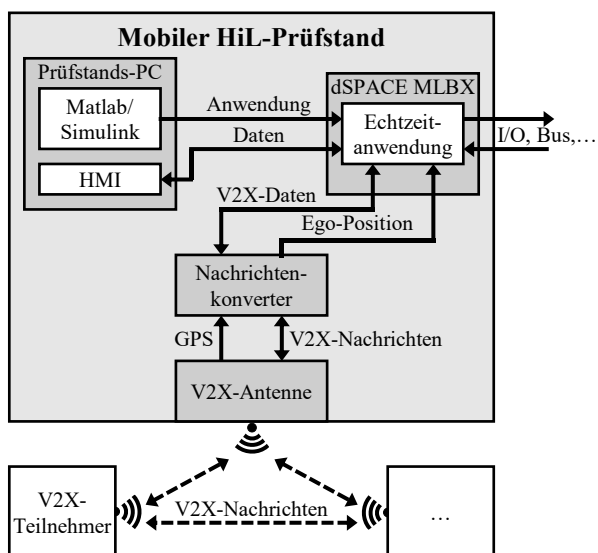


Abbildung 3: Struktur des mobilen HiL-Prüfstands zur Zielführung und V2X-Kommunikation [20].

3 Konzeption der Netzwerk- und Cloudinfrastruktur

Dieser Abschnitt stellt die Konzeption der Netzwerk- und Cloudinfrastruktur dar.

3.1 Definition des Anwendungsbereichs

Als Anwendungsbereich werden urbane Umgebungen betrachtet, da diese deutlich komplexere Gegebenheiten aber auch weitreichendere Optimierungspotentiale in der Zielführung verglichen mit außerstädtischen Bereichen (Land- und Bundesstraßen, Autobahnen) aufweisen:

- Geringere Reichweite der DSRC aufgrund dichter Bebauung, dichten Verkehrs und weiterer Störeinflüsse durch Funknetzwerke und andere Strahlungsquellen.
- Dichteres Straßennetz mit mehr Freiheitsgraden beim Auffinden einer Route und dem Planen von Alternativrouten bei Störungen (z. B. Baustellen, Unfälle).
- Häufigere Störungen durch höheres Verkehrsaufkommen und erhöhte Unfallwahrscheinlichkeit. Nach [21] fanden 2019 in Deutschland ca. 64 % aller Unfälle mit Personenschäden in geschlossenen Ortschaften statt.
- Höhere Anzahl von Kreuzungen, die durch RSUs ausgerüstet und so in ein ITS integriert werden können.

Eine Übertragung der dargestellten Forschungstätigkeit auf außerstädtische Bereiche ist prinzipiell möglich, wird in diesem Beitrag allerdings nicht weiter betrachtet.

3.2 Ableiten von Anforderungen

Basierend auf der Analyse des Standes des Wissens (Abschnitt 1) und dem geplanten Anwendungsbereich (Abschnitt 3.1) werden folgende Anforderungen an die Netzwerkinfrastruktur zur cloudbasierten V2X-Kommunikation und dynamischen Zielführung gestellt:

- Ermöglichen der Kommunikation zwischen zwei Verkehrsteilnehmern, auch wenn die Distanz für die direkte DSRC zu hoch ist.
- Datenverarbeitung mit maximaler Latenz von 5 ms, um die Kommunikation und Nutzung der Daten in Echtzeit zu ermöglichen.
- Unterstützung von DSRC-Nachrichten nach ETSI-Standards (z.B. CAM und DENM).
- Konvertieren von DSRC-Nachrichten, sodass diese in die dritte Schicht einer LDM integriert und für die dynamische Zielführung genutzt werden können.
- Möglichkeit zur Anbindung von Systemen, die nicht explizit für die V2X-Kommunikation vorgesehen

sind. Ein Beispiel stellen Echtzeit-Verkehrssimulationen auf HiL-Prüfständen dar, die für den Funktionstest eines Verkehrsteilnehmers genutzt werden könnten.

3.3 Netzwerkinfrastruktur zur cloudbasierten V2X-Kommunikation

Aus den Anforderungen wird die in Abbildung 4 dargestellte Netzwerkinfrastruktur zur cloudbasierten V2X-Kommunikation abgeleitet, welche dem Prinzip des MEC entspricht. Grundlage bilden Verkehrsteilnehmer, die per DSRC mit einer RSU kommunizieren. Die RSU stellt ein Gateway bzw. einen Edge dar und konvertiert die erhaltenen V2X-Nachrichten, um sie per IP-Netzwerkverbindung (IEEE 802.3) an eine zentrale Cloud zu senden. Diese verarbeitet und speichert die erhaltenen Nachrichten und verteilt sie an die anderen Gateways im Netzwerk, welche die Nachrichten per DSRC wiederum an die in Reichweite befindlichen Verkehrsteilnehmer verteilen. Aufgrund der hohen Reichweite und Erweiterbarkeit der IP-Netzwerkverbindung wird die Reichweite der DSRC signifikant gesteigert, sodass diese auch zur dynamischen Zielführung genutzt werden kann.

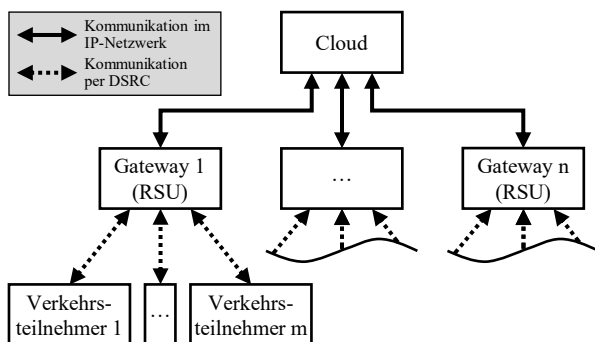


Abbildung 4: Netzwerkinfrastruktur zur cloudbasierten V2X-Kommunikation und dynamischen Zielführung basierend auf dem Prinzip des MEC.

4 Entwurf der Netzwerk- und Cloudinfrastruktur

Der Entwurf der Netzwerk- und Cloudinfrastruktur umfasst die zentrale Cloud (Abschnitt 4.1) sowie die Gateways zwischen Cloud und Verkehrsteilnehmern (Abschnitt 4.2).

4.1 Zentrale Cloud

Die zentrale Cloud ist eine modulare Software, die als transparenter Broker für heterogene Nachrichten von verschiedenen Endpunkten bzw. Gateways auftritt. Mithilfe

der Cloud sollen Daten aus verschiedenen Quellen mit ggf. unterschiedlichem Format und Inhalt zentralisiert erfasst, gespeichert und verteilt werden. Für eine flexible Erweiterbarkeit und die Unabhängigkeit von den Quellen arbeitet die Cloud mit einer einheitlichen Zwischendarstellungsschicht der Nachrichten (Intermediate Representation – IR), sodass unterschiedliche Formate und Inhalte nicht von Relevanz sind. Die Überführung von Nachrichten in die IR und umgekehrt erfolgt durch die Gateways (vgl. Abschnitt 4.2).

Die Cloud verfügt über einen Broker und eine Datenbank. Der Broker koordiniert die Kommunikation zwischen allen Gateways. Er verteilt eingehende IR-Nachrichten nach dem Publish-Subscribe-Verfahren zur Speicherung an die Datenbank und andere Gateways. Nach Bedarf können Nachrichten aus der Datenbank angefragt und erneut verteilt werden.

Der modulare Aufbau der Cloud ermöglicht die Erweiterung um zusätzliche Module bspw. zur Verarbeitung großer Datenmengen oder zur Durchführung komplexer Berechnungen.

4.2 Gateway zwischen Verkehrsteilnehmern und zentraler Cloud

In Abbildung 5 ist schematisch die Schichtenarchitektur eines Gateways bzw. einer RSU, angelehnt an [22] und [23], dargestellt. Ziel ist die Konvertierung von V2X-Nachrichten in IR-Nachrichten und umgekehrt. Neben Funktionen für Netzwerkmanagement und -sicherheit verfügt diese Architektur über vier wesentliche Layer.

Auf dem Access Layer (OSI-Schichten 1 und 2) liegt die WLAN-Kommunikation nach IEEE 802.11p zu den Verkehrsteilnehmern (Austausch von V2X-Nachrichten) und die IP-Netzwerkverbindung nach IEEE 802.3 zur zentralen Cloud (Austausch von IR-Nachrichten). Auf dem Netzwerk & Transportation Layer (OSI-Schichten 3 und 4) erfolgt die Konvertierung der Daten abhängig vom gewählten Kommunikationsprotokoll. Bei diesen handelt es sich um das Basic Transportation Protocol (BTP) für V2X-Nachrichten und UDP (oder auch TCP) für IR-Nachrichten. Auf dem Facility Layer (OSI-Schichten 5 und 6) erfolgt die Verarbeitung der Nachrichten in Abhängigkeit ihres Typs. So werden CAM bspw. durch den CA Service und DENM durch den DEN Service verarbeitet. Auf dem Application Layer (OSI-Schicht 7) liegt die Converter Application, welche eine Transformation zwischen V2X- und IR-Nachricht durchführt und somit die Schnittstelle zwischen den beiden

Kommunikationswegen bildet. Die Converter Application ist spezifisch auf den Endpunkt der Kommunikation angepasst und kann beliebige Informationen, bspw. aus Verkehrssimulationen, in die IR überführen.

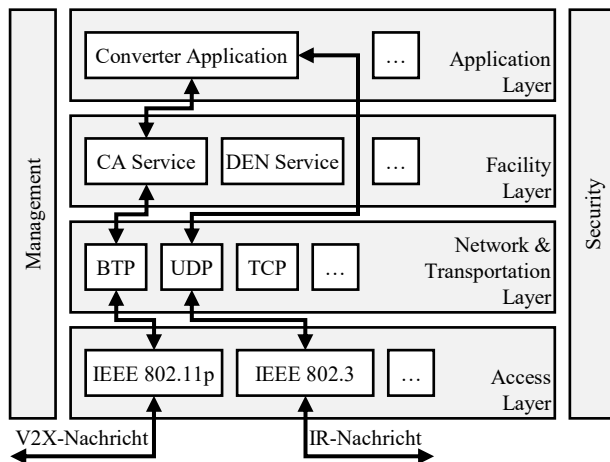


Abbildung 5: Schematische Darstellung der Schichtenarchitektur eines Gateways.

5 Erweiterung der dyn. Zielführung um die cloudbasierte V2X-Kommunikation

Dieser Abschnitt stellt die grundlegende Funktionsstruktur der dynamischen Zielführung und die Integration der V2X-Nachrichten in das Kartenmaterial der LDM dar.

In Abbildung 6 ist die Funktionsstruktur der dynamischen Zielführung dargestellt. Kern der Zielführung ist das, u. a. um Höhendaten der NASA Shuttle Radar Topography Mission ergänzte [24], statische Kartenmaterial der OpenStreetMap [25]. Die Informationen aus der Ermittlung von Fahrzeugzuständen, der Umfeldperzeption und der V2X-Kommunikation werden genutzt, um das statische Kartenmaterial der LDM zu aktualisieren und es um dynamische Informationen auf den Type3/4-Schichten zu erweitern. Des Weiteren werden GPS-Daten aus der Zustandserfassung sowie die Umfeldperzeption mit SLAM-Verfahren zur Selbstlokalisierung genutzt, um den Startpunkt der Zielführung zu bestimmen. Basierend auf der dynamischen Karte, der Ego-Position und dem Fahrtziel wird durch die Zielführung eine bzgl. Fahrdauer, Distanz oder Energieverbrauch optimierte Route ermittelt. Die Zielführung basiert auf dem Dijkstra-Algorithmus ergänzt um eine Heuristik zur Reduktion des Rechenaufwandes. Weiterführende Informationen zur dynamischen Zielführung sind in [26] dargestellt.

Die Kartenaktualisierung erfolgt durch temporäre Anpassung der zu den V2X-Nachrichten gehörenden Kantenbewertungen im Graphen auf der Type3-Schicht der LDM. Zunächst wird aus der GPS-Position der V2X-Nachricht ermittelt, welcher Kante die Information zuzuweisen ist. Anschließend erfolgt die Änderung der Kantengewichte je nach Art der Information. Tritt bspw. eine vollständige Straßensperrung infolge eines Unfalls auf, werden für die entsprechende Kante sowohl Distanz als auch Dauer und Energieverbrauch auf unendlich gesetzt, sodass die Kante nicht mehr Bestandteil der optimalen Route sein kann. Tritt hingegen eine Verzögerung durch stockenden Verkehr an einer Baustelle auf, werden nur Dauer und Energieverbrauch der Kante geändert, da sich die Streckenlänge nicht ändert.

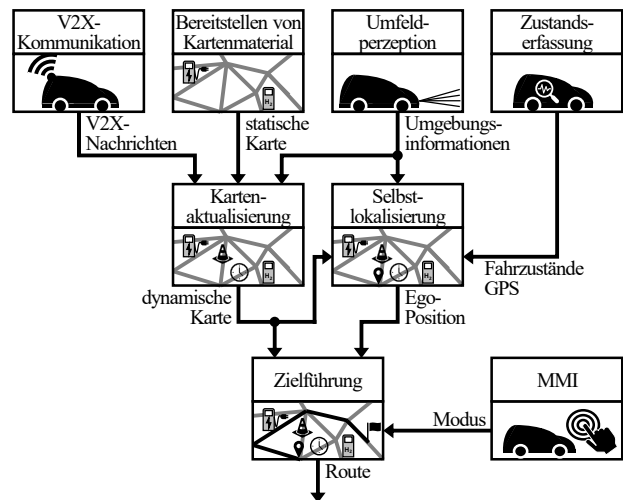


Abbildung 6: Funktionsstruktur der dyn. Zielführung.

6 Verifikation und Validierung

Zur Verifikation und Validierung wird zunächst eine Pilotanwendung definiert (Abschnitt 6.1). Daraufhin erfolgt die Auswertung der in dieser Pilotanwendung durchgeführten Echtzeitsimulationen (Abschnitt 6.2).

6.1 Definition der Pilotanwendung

Die zur Verifikation und Validierung betrachtete Pilotanwendung im urbanen Umfeld der Ostfalia Hochschule in Wolfenbüttel ist in Abbildung 7 dargestellt.

Zur HiL-Simulation unter Echtzeitbedingungen werden verschiedene Systeme verwendet, welche informationstechnisch miteinander gekoppelt sind. Die dynamische Zielführung des Ego-Fahrzeugs (vgl. Abschnitt 5) zur Ermittlung der optimalen Route wird auf dem mobilen HiL-Prüfstand (vgl. Abschnitt 2) ausgeführt. Mittels

DSRC kommuniziert der mobile HiL-Prüfstand mit einer RSU, welche an ein IP-Netzwerk angebunden ist. Über diese Netzwerkverbindung kommuniziert die RSU mit der Cloud, welche wiederum auf einem zentralen Server ausgeführt wird. An die Cloud ist per Netzwerkverbindung zudem der V2X-Umfeldgenerator waveBEE®plus (vgl. Abschnitt 2) angebunden, welcher in der definierten Pilotanwendung die V2X-Nachrichten aller Verkehrsteilnehmer mit Ausnahme des Ego-Fahrzeugs bereitstellt. Diese werden mithilfe der Cloudinfrastruktur an das Ego-Fahrzeug weitergeleitet.

In der Pilotanwendung soll ein V2X-fähiges Fahrzeug von Campus A zu Campus B fahren. Auf der kürzesten Route befindet sich eine Verzögerung von 5 min durch eine kurzfristig eingerichtete Baustelle. In der HiL-Simulation werden zwei exemplarische Szenarien betrachtet: Im ersten Szenario (Referenz) ist die zuvor beschriebene Infrastruktur nicht vorhanden, sodass lediglich die normale DSRC genutzt werden kann. Die effektive Reichweite von ca. 130 m, welche im Umfeld der Ostfalia mit dem mobilen HiL-Prüfstand ermittelt wurde, ist nicht ausreichend für eine frühzeitige Neuberechnung der Route. Im zweiten Szenario ist zusätzlich die cloudbasierte V2X-Kommunikation verfügbar.

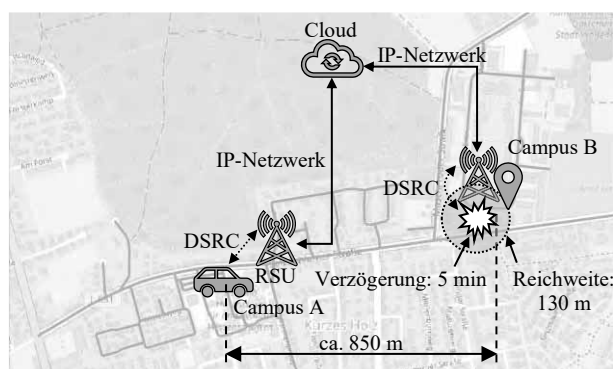


Abbildung 7: Urbane Pilotanwendung an der Ostfalia in Wolfenbüttel mit Infrastruktur zur cloudbasierten V2X-Kommunikation [27].

6.2 Ergebnisse der Echtzeitsimulationen

In diesem Abschnitt werden die Ergebnisse der Echtzeitsimulationen dargestellt. Diese umfassen die Latenzen der Verarbeitung durch die Cloudinfrastruktur und die Ergebnisse der dynamischen Zielführung.

Die Latenz wird zwischen dem Zeitpunkt des Nachrichteneingangs in der Cloud bis zur abgeschlossenen Datenverarbeitung in der Cloud anhand von Zeitstempeln gemessen. Es werden zwei Szenarien mit unter-

schiedlicher Systemauslastung betrachtet. Im ersten Szenario ist das Linux-System, auf dem die Cloud ausgeführt wird, normal ausgelastet. Im zweiten Szenario wird mit weiteren Threads eine maximale Systemauslastung hervorgerufen, um die Latenz bei dieser zu untersuchen. Tabelle 1 beinhaltet die Ergebnisse der Latenzanalyse. Unter normaler Systemauslastung werden durchschnittliche und maximale Latenzzeiten erzielt, welche deutlich unterhalb des Grenzwerts von 5 ms liegen und eine hinreichend schnelle Verarbeitung garantieren. Unter erhöhter Systemlast wird ein deutliches Überschreiten des Grenzwertes in über 80 % der Fälle festgestellt. Hieraus folgt, dass unter dieser Systemlast unzulässig hohe Verzögerungen bei der Datenverarbeitung auftreten können. Es ist allerdings anzumerken, dass die (künstlich hervorgerufene) erhöhte Systemlast in der finalen Anwendung auf einem eigens für die Cloud vorgesehenen Rechner durch entsprechende Sicherheits- und Priorisierungsmaßnahmen vermieden wird.

Weiterhin wurde messtechnisch erfasst, dass die Latenz der Nachrichtenübertragung von einer RSU zu einer anderen mittels IP-Netzwerkkommunikation über die Cloud (inkl. Verarbeitungszeit der Cloud) durchschnittlich unterhalb von 2 ms liegt. Verglichen mit den gemessenen Latenzen der DSRC (160 bis 250 ms) sind diese vernachlässigbar.

	Normale Systemlast	Erhöhte Systemlast
t_{\max} in ms	1,42	56,67
t_{avg} in ms	0,04	23,09
$t \geq 5$ ms	0,00 %	85,02 %

Tabelle 1: Latenzzeiten der Nachrichtenverarbeitung durch die Cloud.

Abbildung 8 zeigt die Ergebnisse der dynamischen Zielführung. Im ersten Szenario (Abbildung 8a) mit normaler DSRC wird die ohne Baustelle beste Route gewählt, welche allerdings aufgrund der fehlenden Benachrichtigung durch die Baustelle führt. Es ergibt sich inklusive fünfminütiger Verzögerung eine Fahrtdauer von 7,67 min. Im zweiten Szenario (Abbildung 8b) ist die cloudbasierte V2X-Kommunikation verfügbar und es wird frühzeitig eine Route um die Baustelle herum gewählt. Das Ziel wird in 5,94 min und somit 1,73 min schneller als in Szenario 1 erreicht.



a: Optimale Route ohne cloudbasierte V2X-Kommunikation.



b: Optimale Route mit cloudbasierter V2X-Kommunikation.

Abbildung 8: Gegenüberstellung der gewählten Routen von Campus A zu Campus B ohne (a) und mit (b) cloudbasierter V2X-Kommunikation.

Die dargestellten Ergebnisse zeigen zum einen die Funktionsfähigkeit der Cloudinfrastruktur mit einer hinreichend kurzen Latenz von weniger als 5 ms, sodass kein signifikanter Aufschlag auf die Latenz der DSRC vorliegt. Zum anderen verifizieren und validieren sie die Integration der cloudbasierten V2X-Kommunikation in die dynamische Zielführung und demonstrieren den Mehrwert durch das Auffinden besserer Routen.

7 Resümee und Ausblick

In diesem Beitrag wurde eine auf MEC basierende Netzwerk- und Cloudinfrastruktur dargestellt, welche eine signifikante Reichweitenerhöhung der V2X-Kommunikation mittels DSRC erzielt. RSUs dienen als Schnittstelle zwischen Verkehrsteilnehmern und einer zentralen Cloud. Die Kommunikation zwischen RSUs und Verkehrsteilnehmern erfolgt mittels DSRC und zwischen RSUs und zentraler Cloud in IP-Netzwerken.

Zudem wurde in der dynamischen Zielführung eine Schnittstelle definiert, mit welcher die V2X-Nachrichten als dynamische Informationen in das Kartenmaterial der LDM integriert werden können. Basierend auf dem aktualisierten Kartenmaterial ist eine verbesserte Zielführung mit frühzeitiger Alternativroutenplanung möglich, wel-

che in Echtzeitsimulationen mithilfe einer Pilotanwendung am Standort Wolfenbüttel der Ostfalia nachgewiesen wurde.

Im Rahmen des Niedersächsischen Zukunftslabors Mobilität wird eine Weiterführung der dargestellten Forschungstätigkeiten angestrebt. Insbesondere soll das Potential der neu verfügbaren V2X-Informationen für eine kooperative Zielführung mehrerer Verkehrsteilnehmer zur Reduktion oder Vermeidung von Staus untersucht werden. Des Weiteren soll eine Integration der C-V2X-Kommunikation zur dynamischen Zielführung durchgeführt werden, da diese sich langfristig gegen die WLAN-basierte DSRC durchzusetzen scheint (vgl. Entzug der Frequenzbänder für DSRC in den USA [28]). In Hinblick auf diese Integration wurde die Schnittstelle zur Informationseinbindung in die LDM so definiert, dass eine nachträgliche Änderung der Kommunikationstechnologie mit geringem Aufwand möglich ist.

Danksagung

Dieser Beitrag wurde im Rahmen des Verbundprojekts *Zukünftige Fahrzeugtechnologien im Open Region Lab* durch das Niedersächsische Ministerium für Wissenschaft und Kultur sowie die VolkswagenStiftung unter dem Förderkennzeichen VWZN3236 gefördert.



Niedersächsisches Ministerium
für Wissenschaft und Kultur



VolkswagenStiftung

Referenzen

- [1] Scherler S, Pramme F, Liu-Henke X, Bicker G. iREX 4.0 – A contribution to a predictive, energy-optimal drive of Autonomous Electric Vehicles equipped with Range Extender by means of Cross-linking and Digitization. 16th Symposium on Hybrid and Electric Vehicles, Braunschweig, Germany, 20. - 21. Februar, 2019.
- [2] Velasco-Hernandez G, Caicedo-Bravo E, Barry J, Walsh J. Intersection Management Systems and Internet of Things: A Review. IEEE 16th Int. Conference on Intelligent Computer Communication and Processing, 2020.
- [3] Napolitano A, Cecchetti G, Giannone F, Ruscelli A L, Civerchia F, Kondepu K, Valcarengi L, Castoldi P. Implementation of a MEC-based Vulnerable Road User Warning System. AEIT Int. Conference of Electrical and Electronic Technologies for Automotive, 2019.
- [4] Tong W, Hussain A, Bo W X, Maharjan S. Artificial Intelligence for Vehicle-to-Everything: A Survey. IEEE Access, Vol 7, S. 10823 – 10843, Januar 2019.
- [5] Naik G, Choudhury B, Park J-M, IEEE 802.11bd & 5G NR V2X: Evolution of Radio Access Technologies for V2X Communications, IEEE Access, Vol. 7, S. 70169 - 70184, Juni 2019.

- [6] Lembke R. DSRC und C-V2X im Vergleich. Automatisiertes Fahren, SpringerProfessional, <https://www.springerprofessional.de/automatisiertes-fahren/automobil-elektronik---software/dsrc-und-c-v2x-im-vergleich/15476434>. Stand: 06.03.2018.
- [7] Mannoni V, Berg V, Sesia S, Perraud E. A Comparison of the V2X Communication Systems: ITS-G5 and C-V2X. IEEE Vehicular Technology Conference (VTC), Kuala-Lumpur, Malaysia, April, 2019.
- [8] Perry F. Overview of DSRC Messages and Performance Requirements. UFTI DSRC and Other Communication Options for Transportation Connectivity Workshop, 03. Mai, 2017.
- [9] ETSI. ETSI TS 102 637-2, Intelligent Transport Systems (ITS); Vehicular Communications; Basic Set of Applications; Part 2: Specification of Cooperative Awareness Basic Service, März 2011.
- [10] ETSI. ETSI TS 102 637-3, Intelligent Transport Systems (ITS); Vehicular Communications; Basic Set of Applications; Part 3: Specifications of Decentralized Environmental Notification Basic Service, November 2014.
- [11] Masood A, Lakew D S, Cho S. Security and Privacy Challenges in Connected Vehicular Cloud Computing. IEEE Communications Surveys & Tutorials. Vol. 22, Nr. 4, 2020.
- [12] Nkenyereye L, Nkenyereye L, Islam S M R, Kerrache C A, Abdullah-Al-Wadud M, Alamri A. Software Defined Network-Based Multi-Access Edge Framework for Vehicular Networks. IEEE Access, Vol. 8, S. 4220 - 4234, 2020.
- [13] Dijkstra E W. A Note on Two Problems in Connexion with Graphs. Numerische Mathematik. S. 269 - 271, 1959.
- [14] Bast H et al. Route Planning in Transportation Networks. Lecture Notes in Computer Science, Vol. 9220, S. 19 - 80, 2016.
- [15] Dorigo M et al. Ant algorithms for discrete optimization. Artificial Life, Vol. 5, Nr. 2, S. 137 - 172, 1999.
- [16] Yu J, Yu W, Gu J. Online Vehicle Routing With Neural Combinatorial Optimization and Deep Reinforcement Learning. IEEE Transactions on Intelligent Transportation Systems, Vol. 20, Nr. 10, Oktober 2019.
- [17] Schmidt M, Kwella B, Pieth N. TPEG löst TMC ab – Eine neue Generation der Navigation. ATZelektronik, Vol. 4, Nr. 6, 2009.
- [18] Shimada H, Yamaguchi A, Takada H, Sato K. Implementation and Evaluation of Local Dynamic Map in Safety Driving Systems. Journal of Transportation Technologies, Vol. 5, S. 102 - 112, 2015
- [19] Liu-Henke X, Duym S. Modellgestützte Funktionsabsicherung des vernetzten mechatronischen Kraftfahrzeugs. VDI-Tagung Mechatronik, Wiesloch, 01./02. Juli, 2005.
- [20] Scherler S, Liu-Henke X. Conception and Realization of a Mobile HiL Test Bench for V2X Communication. IEEE 91st Vehicular Technology Conference (VTC2020-Spring), Virtual, May 25 - July 31, 2020.
- [21] Deutscher Verkehrssicherheitsrat e.V. Unfallstatistik. www.dvr.de/service/unfallstatistik, Stand: 10.02.2021.
- [22] ETSI. ETSI TS 302 665, Intelligent Transport Systems (ITS); Communications Architecture, September 2010.
- [23] Hobert L, Festag A, Llatser I, Altomare L, Visintainer F, Kovacs A. Enhancements of V2X Communication in Support of Cooperative Autonomous Driving. IEEE Communication Magazine, S. 64 - 70, Dezember, 2015.
- [24] Farr T G. The Shuttle Radar Topography Mission. Reviews of Geophysics, 45, RG2004, 2007
- [25] OpenStreetMap Foundation. OpenStreetMap. www.openstreetmap.org, Stand: 05.02.2021
- [26] Scherler S, Liu-Henke X. Intelligente Zielführung elektrischer Fahrzeuge mit Brennstoffzelle als Range Extender in vernetzten Verkehrssystemen. 25. Symposium Simulationstechnik, Virtual, 14./15. Oktober, 2020.
- [27] Pramme F, Scherler S, Kelm L R, Frank L, Bikker G, Liu-Henke X. Zukunft der Fahrzeugkommunikation - Fährst du schon oder stehst du noch? Ostfalia Campus als Testfeld für V2X-Kommunikation und automatisierte Fahrfunktionen. Embedded Software Engineering Kongress, Sindelfingen, Germany, 30. November - 04. Dezember, 2020.
- [28] Zivadinovic D. Autovernetzung: USA schicken 11p-WLAN in Rente. www.heise.de/news/Autovernetzung-USA-schicken-11p-WLAN-in-Rente-4966174.html, Stand: 21.11.2020

Simulation Scenarios for Testing Airborne Cyber-Physical Systems

Siddhartha Gupta^{1*}, Umut Durak², Sven Hartmann¹

¹Institute of Informatics, TU Clausthal, Julius-Albert-Str. 4,
38678 Clausthal-Zellerfeld, Germany; *siddhartha.gupta@tu-clausthal.de

²Institute of Flight Systems, German Aerospace Center (DLR), Lilienthalplatz 7, 38108 Braunschweig, Germany

Abstract. Aircraft are extremely complex Cyber-Physical Systems (CPS) with many component interactions to control the flight and provide rich functionality. The rise in the utilization of modelling and simulation based approaches for CPS development is phenomenal. The Model-Based Design (MBD) brought the system models in the center of the development process. Simulation-based verification established the practice for executing system models as the native mechanisms of testing. Model-in-the-loop, software-in-the-loop and hardware-in-the-loop, commonly named as x-in-the-loop testing became the main steps of simulation-based verification. Feedback loops make the controller and plant interaction crucial in CPS. In order to create an execution of a credible test case, the plant and controller models are simulated in a scenario in a closed-loop fashion. This is called scenario testing. Simulation scenario is a specification for the initial and terminal conditions, significant events and the environment as well as the major entities, their capabilities, behavior and interactions over time. Scenario development is an extensive process beginning with the stakeholders' descriptions of the scenario and finishing with the generation of the corresponding executable specifications. This paper discusses the development and use of scenarios for testing airborne CPS.

Introduction

Cyber-Physical Systems (CPS) are integration of computation and physical processes [1]. An aircraft is an extremely complex CPS, whose physical world includes mechanical aircraft parts, pilots, the physical airspace, the terrain and all other man-made systems associated with aircraft, and whose cyber world encompasses avionics systems, sensors and actuators as well as the networking elements such as data buses. CPS technologies are vital for the future of aeronautics.

Simulations play a major role in the engineering process as they allow engineers to test designs and prototypes without spending excessive resources on construction and manufacturing. A computer simulation is relatively easier to deploy. The cost of failure is minimal,

encouraging experimentation and creative thinking.

Scenarios are important artifacts in the simulation engineering process [2]. A simulation study begins with a description of a scenario and ends with a successful simulation execution [3]. They play a key role in designing cost-effective methods for testing existing systems as well as designing new concepts on them. Elements of a scenario generally include the initial states, the environment, entities, and their capabilities, associated events, its timelines as well as the termination conditions. Scenario development is viewed as the transformation of operational scenarios (defined using natural language) to conceptual scenarios (conforming to a metamodel), and eventually to executable scenarios (specified using a scenario specification language) [4]. However, there was no common understanding of the elements and the structure of these elements in a scenario. This leads to deficiencies in communication and presenting test scenarios, sharing and reusing them, verifying and validating them, and prevents the development of common infrastructures for scenario development and execution.

Karmokar et al. [4] proposed System Entity Structures (SES) for metamodeling and its derivative structure, a Pruned Entity Structure (PES) which represents a particular scenario as effective components of an open simulation infrastructure. SES is a high-level ontology which was introduced for knowledge representation of decomposition, taxonomy and coupling of systems [5]. It represents elements of a system and their relationships in hierarchical and axiomatic manner. This helps in modelling all possible elements in form of a scenario metamodel. To extract a particular scenario from this metamodel, an operation called pruning is performed, which results in a decision free tree called a PES.

Durak et al. [6] promoted XML Schema as the computational representation of scenario elements in SES where each scenario in PES is an XML document. This

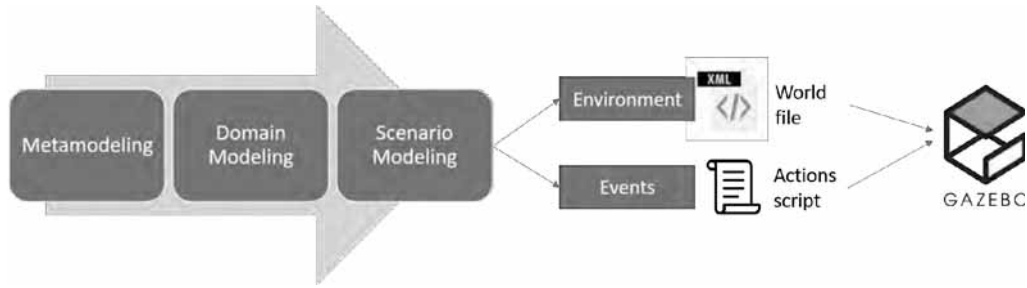


Figure 1: Simplified Workflow for Scenario-based Testing

provides a formal bases not only for the syntax but also for the semantics of scenario description thereby aids sharing and reuse of scenarios.

An important requirement of a scenario-based testing is executable characteristics of the scenarios specification. The user should be able to design a specific scenario from a scenario metamodel and execute it with a selected a simulator. This will allow the user to test the behaviour using scenarios. One such commonly used open source simulator is Gazebo [7], open source robotics simulator . It is generally used together with Robot Operating System (ROS) [8], which is a collection of tools and libraries which helps simplifying the task of building a robot system. Gazebo is a simulation tool which helps to simulate robots in different environments and supports the same communication infrastructure as ROS. ROS programs use plugins to interact with Gazebo robot models within the simulator. Gazebo and ROS have been used quite intensively in the last decade for developing and testing drones. Examples include [9, 10, 11, 12].

In this paper, we introduce very briefly a scenario-based testing approach where we use metamodeling to describe a language to model all the elements of a scenarios using SES, then pruning for scenario modeling. The corresponding infrastructure provides all the required components starting from scenario description to transformation to an executable scenario and the execution of this scenario in Gazebo. An sample use case is intended based on [12] where the components of the Service Oriented Mission System for Autonomous Drones (SOMSAD) will be tested on a Gazebo environment using the ROS infrastructure.

1 Scenario-Based Testing

A simplified representation of the scenario-based testing workflow is presented in Figure 1 At the highest level, it can be broken down into three aspects each described below.

1. **Scenario Modeling** - This includes three steps. We use SES as the metamodel following the footsteps of [4]. Then we model all the possible elements of the scenarios in consideration using SES into a scenario metamodel. Then we prune specific elements to come up with a specific scenario PES. This scenario model is in the XML format. The two main parts of this XML file are the initial setup of entities referred to as the environment, and the events taking place on the entities.
2. **Environment Setup** - We use scripts to extract the Environment section from the scenario model. It is then mapped to World file of the Gazebo simulator. The World file is of Simulation Description Format (SDF) , which is an XML format that describes objects and environments for robot simulators, visualization, and control. This World file can be used to launch the Gazebo simulator with the initial environmental setup.
3. **Events** - We use action scripts to extract events out of the scenario model and execute events in the Gazebo environment. The scripts send the events to the SOMSAD system using the REST interface which gets executed in Gazebo.

The implementation of the workflow in Figure 1 requires an infrastructure, tools and the intermediate scripts or interfaces for the tools to be able to pass data to each

other. The major tools required for this workflow already exist and needs to be extended to able to provide the functionality required for the workflow.

The tools used in the workflow are SES Editor, PES Editor, Gazebo and SOMSAD.

- **SES Editor** - A GUI tool to build SES models using all the SES elements and drag and drop options. It supports an ability to export XML files that can be imported by the PES editor [4].
- **PES Editor** - A GUI tool to import SES models and prune the model for a selection free tree [4].
- **Gazebo** - An open-source robotics simulator [7].
- **SOMSAD** - A service oriented drone mission system [12].

2 Outlook

The scenario-based testing approach introduced in the previous section will be supported by an open scenario infrastructure. The infrastructure will be extended to support interfaces with a variety of simulators. The first use case will be testing collaborative lifting scenarios where drones and cranes working together in a construction site. The future work includes scenario-based testing with various other simulation environments.

References

- [1] Lee EA. Cyber physical systems: Design challenges. In: *2008 11th IEEE International Symposium on Object and Component-Oriented Real-Time Distributed Computing (ISORC)*. IEEE. 2008; pp. 363–369.
- [2] 1730TM-2010 IS. IEEE Recommended Practice for Distributed Simulation Engineering and Execution Process (DSEEP). 2011;.
- [3] Jafer S, Durak U. Tackling the complexity of simulation scenario development in aviation. In: *Proceedings of the Symposium on Modeling and Simulation of Complexity in Intelligent, Adaptive and Autonomous Systems*. 2017; pp. 1–10.
- [4] Chandra Karmokar B, Durak U, Jafer S, Chhaya BN, Hartmann S. Tools for Scenario Development Using System Entity Structures. In: *AIAA Scitech 2019 Forum*. 2019; p. 1712.
- [5] Kim TG, Lee C, Christensen ER, Zeigler BP. System entity structuring and model base management. *IEEE transactions on systems, man, and cybernetics*. 1990;20(5):1013–1024.
- [6] Durak U, Jafer S, Wittman R, Mittal S, Hartmann S, Zeigler BP. Computational representation for a simulation scenario definition language. In: *2018 AIAA Modeling and Simulation Technologies Conference*. 2018; p. 1398.
- [7] Koenig N, Howard A. Design and use paradigms for gazebo, an open-source multi-robot simulator. In: *2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)(IEEE Cat. No. 04CH37566)*, vol. 3. IEEE. 2004; pp. 2149–2154.
- [8] Quigley M, Conley K, Gerkey B, Faust J, Foote T, Leibs J, Wheeler R, Ng AY. ROS: an open-source Robot Operating System. In: *ICRA workshop on open source software*, vol. 3. Kobe, Japan. 2009; p. 5.
- [9] Meyer J, Sendobry A, Kohlbrecher S, Klingauf U, Von Stryk O. Comprehensive simulation of quadrotor uavs using ros and gazebo. In: *International conference on simulation, modeling, and programming for autonomous robots*. Springer. 2012; pp. 400–411.
- [10] Bernardeschi C, Fagiolini A, Palmieri M, Scrima G, Sofia F. Ros/gazebo based simulation of cooperative uavs. In: *International Conference on Modelling and Simulation for Autonomous Systems*. Springer. 2018; pp. 321–334.
- [11] Sciortino C, Fagiolini A. ROS/Gazebo-Based Simulation of Quadcopter Aircrafts. In: *2018 IEEE 4th International Forum on Research and Technology for Society and Industry (RTSI)*. IEEE. 2018; pp. 1–6.
- [12] Gupta S, Durak U. RESTful Software Architecture for ROS-based Onboard Mission System for Drones. In: *AIAA Scitech 2020 Forum*. 2020; p. 0239.

Modellbasierte Entwicklung des Antriebsmoduls für vernetzte fahrerlose Transportfahrzeuge in einem cyber-physischen Labortestfeld

Sven Jacobitz^{1*}, Marian Göllner¹, Jie Zhang¹, Xiaobo Liu-Henke¹

¹Institut für Mechatronik, Ostfalia Hochschule für angewandte Wissenschaften, Salzdahlumer Str. 46/48, 38302 Wolfenbüttel; *sve.jacobitz@ostfalia.de

Kurzfassung. Zur realitätsnahen Entwicklung und Optimierung autonomer Fahrfunktionen wird an der Ostfalia das cyber-physische Labortestfeld für intelligente Mobilitätsanwendungen aufgebaut. Kern dieses Labortestfeldes sind fahrerlose Transportfahrzeuge (FTF), welche als Mobilitätsplattform eingesetzt werden. Diese sind drahtlos miteinander und mit anderen Akteuren vernetzt, um Informationen, z. B. über Hindernisse oder gefährliche Situationen, austauschen zu können. Dieser Beitrag beschreibt die modellbasierte Entwicklung des Antriebsmoduls der FTF. Es wird das Konzept der FTF vorgestellt und die Informationsverarbeitung strukturiert. Anhand eines identifizierten und verifizierten Modells wird anschließend die Fahrdynamikregelung ausgelegt und mittels Model-in-the-Loop-, Software-in-the-Loop- und Hardware-in-the-Loop-Simulationen verifiziert.

Einleitung

Hoch automatisierte Fahrzeuge bieten das Potenzial, den Straßenverkehr durch Steigerung der Verkehrssicherheit und -effizienz zu revolutionieren [1]. Mit der Strategie automatisiertes und vernetztes Fahren hat die Bundesregierung beschlossen, dass Deutschland hierbei eine Spitzenposition einnehmen soll. Die Einrichtung von Testfeldern zur Erprobung unter realitätsnahen Bedingungen spielt dabei eine besondere Rolle [2]. Im Rahmen des Niedersächsischen Zukunftslabors Mobilität wird daher an der Ostfalia ein cyber-physisches Labortestfeld für intelligente Mobilitätsanwendungen aufgebaut (vgl. Abbildung 1) [3].

Das Labortestfeld bietet eine Plattform, in welcher Mobilitätsanwendungen – skaliert auf Laborgröße – unter Echtzeitbedingungen erprobt und optimiert werden können. Es werden Anwendungen aus verschiedenen Domänen wie dem Personenverkehr, dem Gütertransport in Industrie-4.0-Anlagen und vieles mehr un-

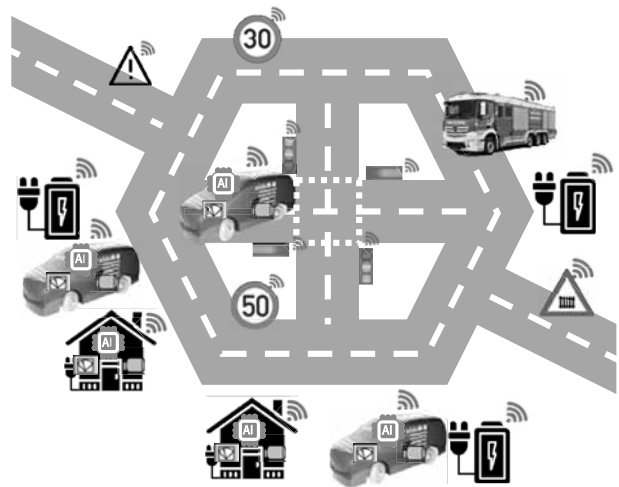


Abbildung 1: Konzept des Labortestfelds für intelligente Mobilitätsanwendungen.

terstützt. Kern des Testfelds sind modular aufgebaute, voll vernetzte Mobilitätsplattformen auf Basis fahrerloser Transportfahrzeuge (FTF). Der folgende Beitrag beschreibt die Entwicklung des Antriebsmoduls einer solchen Mobilitätsplattform.

Der restliche Beitrag ist wie folgt aufgebaut. In Kapitel 1 wird der Stand des Wissens hinsichtlich FTF analysiert. Kapitel 2 befasst sich mit der Entwurfsmethodik, worauf in Kapitel 3 das Konzept des FTF aufgebaut wird. Kapitel 4 befasst sich mit der mechanischen und informationstechnischen Konzeption des Antriebsmoduls, woraufhin in Kapitel 5 ein Simulationsmodell des FTF gebildet wird. Auf Basis dieses Modells erfolgt in Kapitel 6 die Funktionssynthese und in Kapitel 7 die Realisierung des Antriebsmoduls.

1 Stand des Wissens

Bereits Mitte der 50er Jahre des letzten Jahrhunderts wurden fahrerlose Transportsysteme – ein Vorgänger moderner FTF – zum automatisierten Gütertransport in der Industrie eingesetzt [4]. Diese spurgebundenen Transportsysteme waren jedoch wenig flexibel, sodass seit 2010 vernetzte, autonom fahrende, FTF forciert werden [5]. Inzwischen sind hoch automatisierte FTF fester Bestandteil moderner Produktionsanlagen.

Aufgrund der weitreichenden Erfahrungen hinsichtlich Navigation und automatisierten Fahrfunktionen eignen sich FTF hervorragend als Mobilitätsplattform im Labortestfeld. Durch Standardisierung der Schnittstellen [6] lassen sie sich mit geringem Aufwand in Mobilitätssimulationen anderer Domänen integrieren.

Häufig werden FTF mit omnidirektionalen Antrieben ausgeführt [4], um frei und flexibel in räumlich restriktiven Situationen agieren zu können. Hierbei werden z. B. Mecanum-Räder eingesetzt [7]. Für einen möglichst autonomen Gütertransport werden zudem dezentrale Entscheidungsebenen benötigt. Damit in diesem Zusammenhang eine Abstimmung einzelner Systeme untereinander stattfinden kann, ist die Vernetzung unerlässlich. Hierdurch entsteht ein cyber-physisches System (CPS), da die einzelnen physischen Komponenten nun rein über Netzwerke und moderne Informationstechnik miteinander verbunden sind. Die vorhandenen Erfahrungen aus dem Bereich Industrie-4.0 können auf die Domäne Straßenverkehr transferiert werden.

2 Methodik

Die Entwicklung von CPS stellt eine besondere Herausforderung dar [3]. Zur Beherrschung der Komplexität, ist ein systematisches Vorgehen unumgänglich. Aus diesem Grund wird für die Entwicklung intelligenter Mobilitätsanwendungen, welche ein CPS bilden, sowie deren Teilkomponenten die durchgängig modellbasierte, ganzheitliche mechatronische Entwicklungsmethodik angelehnt an [8] angewendet. Im Top-Down-Verfahren erfolgt zunächst eine Modularisierung und Hierarchisierung des Systems zur Beherrschung der Komplexität. Anschließend werden, beginnend mit der untersten Hierarchieebene, die Module ausgelegt und bis hin zum Gesamtsystem integriert. Zur Optimierung und zum Test auf den einzelnen Ebenen werden Model-in-the-Loop (MiL)-, Software-in-the-Loop (SiL)- und Hardware-in-the-Loop (HiL)-Simulationen bereits in

frühen Phasen der Entwicklung durchgeführt. Insbesondere die Optimierung und Absicherung auf höheren Hierarchieebenen des CPS stellt hierbei eine große Herausforderung dar. Die Komplexität und der hohe Vernetzungsgrad erfordert den Einsatz von Prototypen unter realistischen Bedingungen. Hierzu wird das Labortestfeld nach [3] eingesetzt.

Die Auslegung des Antriebsmoduls erfolgt mit der Low-Cost RCP-Entwicklungsplattform LoRra [9]. Unter Einsatz des Open-Source CAE-Werkzeugs Scilab / Xcos und eines kostengünstigen Mikrocontrollers als Echtzeithardware wird die ganzheitliche mechatronische Entwicklungsmethodik, von der Modellbildung bis hin zur Echtzeitrealisierung im Rahmen von HiL-Simulationen, durchgängig umgesetzt.

3 Konzept des FTF

Das FTF ist modular aufgebaut und dient als grundlegende Mobilitätsplattform im Labortestfeld. Abbildung 2 illustriert das Konzept des FTF. Es besteht hauptsächlich aus einem Perzeptionsmodul zur Erfassung der Umgebung mittels Simultaneous Localization and Mapping (SLAM), einem Kommunikationsmodul zur WLAN-basierten Vernetzung, einem Verarbeitungsmodul zur Berechnung intelligenter Fahrfunktionen unter Echtzeitbedingungen sowie einem Antriebs- und Energiemodul, welches Fokus dieses Beitrags ist. Durch diesen modularen Aufbau ist das FTF flexibel einsetzbar und kann als Objekt zur Beantwortung diverser Forschungsfragen genutzt werden.

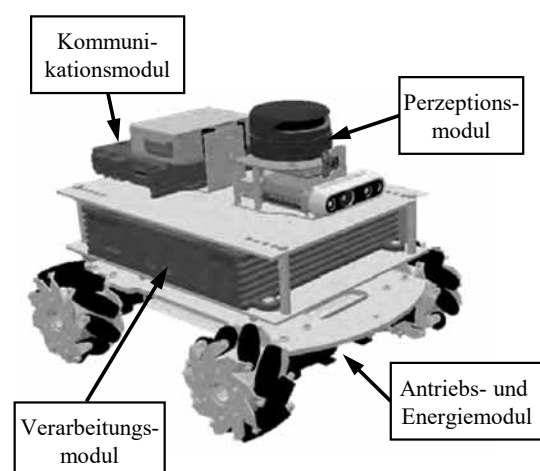


Abbildung 2: Modulares Konzept des FTF.

Die Module des FTF sind durch definiert physikalische und informationstechnische Schnittstellen miteinander verbunden. Dies ermöglicht sowohl die separate, parallele Entwicklung als auch die Variation einzelner Module, ohne Anpassungen am restlichen Aufbau. So kann z. B. für das Verarbeitungsmodul High-Cost-RCP-Hardware in Form einer MicroAutobox aber auch das in Kapitel 2 erwähnte Low-Cost-RCP-System LoRra eingesetzt werden. Abbildung 3 illustriert die Struktur und die Schnittstellen des FTF.

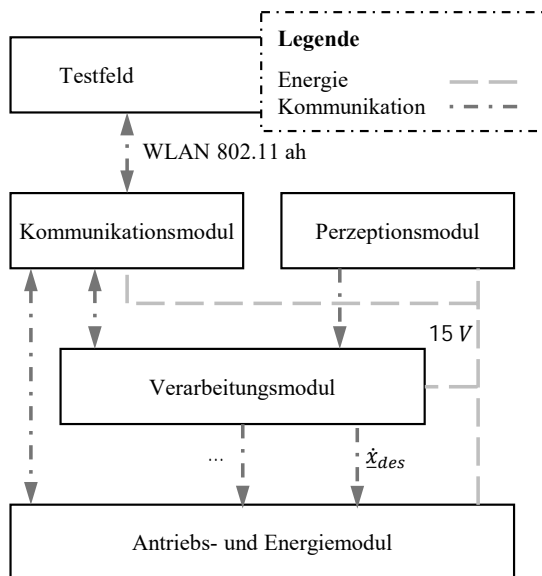


Abbildung 3: Struktur des FTF mit wesentlichen Schnittstellen.

4 Konzeption des Antriebsmoduls

Das Antriebs- und Energiemodul (kurz Antriebsmodul) ist eines der grundlegenden Module des FTF. Auf ihm bauen alle anderen Module auf, sodass es einen Großteil der Trag- und Stützstruktur der Plattform beinhaltet. Es besteht aus mechanischen und informationstechnischen Komponenten, welche übergeordnet drei Aufgaben erfüllen:

- **Energieversorgung:** Das Antriebsmodul versorgt die anderen Module des FTF mit Energie. Untergeordnet werden hierzu Funktionen wie Energiemanagement und Lademanagement benötigt.
- **Sicherheitsmanagement:** Das Antriebsmodul muss unsichere Zustände des FTF erkennen und es

durch entsprechende Aktionen in einen definierten sicheren Zustand versetzen. Dies gilt sowohl für Energietechnische und Informationstechnische Zustände als auch für unsichere Fahrsituationen.

- **Antreiben:** Kernfunktion des Moduls ist der omnidirektionale Antrieb des FTF. Die Aufgabe *antreiben* ist Inhalt dieses Beitrags.

An die mechanischen und informationstechnischen Funktionen zur Erfüllung der Aufgabe *antreiben* werden diverse Anforderungen gestellt. Folgende wesentliche Anforderungen sind neben obligatorischen (z. B. Stabilität und funktionale Sicherheit) im Rahmen dieses Beitrags relevant:

- Omnidirektionaler Antrieb mit einer minimalen Anzahl an Aktoren.
- Gezielte und unabhängige Regelung der Bewegung in Längs-, Quer- und Gierrichtung.
- Vorgabe des Sollgeschwindigkeits-Vektors über das Verarbeitungsmodul oder eine RC-Fernbedienung.
- Stationäre Genauigkeit der Regelung (max. Abweichung 1 %).
- Geringe Überschwingweite der Regelung (Lehrsches Dämpfungsmaß $\zeta > 0,7$).
- Ausnutzung der maximalen Bandbreite unter Berücksichtigung der Stellgrößenbeschränkung.

4.1 Mechanische Konzeption

Um einen omnidirektionalen Antrieb zu realisieren gibt es verschiedene Möglichkeiten, welche meist zusätzliche Aktorik benötigen. Durch den Einsatz von Mecanum-Rädern lässt sich dies vermeiden [10]. Eine gezielte Anordnung der Antriebe in Kombination mit einer entsprechenden Ansteuerung ermöglicht das omnidirektionale Verfahren mit minimaler Aktorik [11].

In Anlehnung an [7] wird aus Stabilitätsgründen eine Topologie mit vier radindividuellen elektrischen Antrieben gewählt. Abbildung 4 illustriert das Antriebskonzept.

4.2 Informationstechnische Konzeption

Wesentliche Schnittstelle des Moduls Antrieb zum Verarbeitungsmodul ist der Geschwindigkeits-Sollwertvektor mit den Geschwindigkeitskomponenten in x-

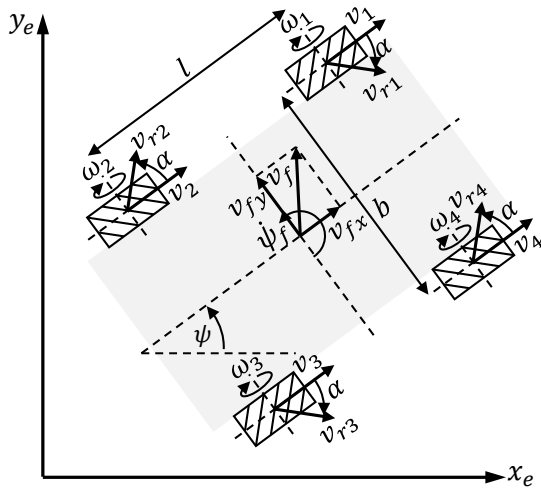


Abbildung 4: Antriebskonzept des FTF.

(v_{fx}) und y-Richtung (v_{fy}) sowie der Giergeschwindigkeit (ψ_f) des Fahrzeugschwerpunktes im Fahrzeug festem Koordinatensystem:

$$\dot{x}_{des} = [v_{fx,des} \quad v_{fy,des} \quad \psi_{f,des}]^T \quad (1)$$

Dieser soll durch entsprechende Ansteuerung der Mecanum-Räder eingeregelt werden. Zudem soll für die Positionierung der FTF im Testfeld noch eine ferngesteuerte Fahrt, unabhängig vom Verarbeitungsmodul, möglich sein. Hierzu wird zwischen den Fahrmodi autonome Fahrt und ferngesteuerte Fahrt unterschieden. Für den Betrieb des Antriebsmodul ergibt sich ein Zustandsautomat mit den vier Zuständen *autonome Fahrt*, *ferngesteuerte Fahrt*, *Stillstand* und *Fehler*.

Die Aufgabe *antreiben* wird durch eine Funktion Antriebsregelung realisiert. Abbildung 5 illustriert vereinfacht die resultierende Struktur. Abhängig vom aktuellen Zustand wird die Quelle für \dot{x}_{des} umgeschaltet. Mittels Entkopplung erfolgt die Bestimmung des quasi modalen Zustandsvektors \dot{x} aus der Odometrie des FTF. Kern der Antriebsregelung ist die Fahrdynamikregelung, welche in Kapitel 6 ausgelegt wird. Sicherheitsfunktionen prüfen in Abhängigkeit des aktuellen Zustands sowie weiteren Randbedingungen, ob die Leistungsausgabe freigegeben wird und ob der geforderte Antriebskraftvektor \underline{F} zulässig ist. Über eine Verkopplungsstruktur erfolgt die Berechnung eines PWM-Signals \underline{u} , mit welchem die Aktoren angesteuert werden.

Durch diese Struktur aus Entkopplung, Fahrdy-

namikregelung und Verkopplung vereinfacht sich die Reglerauslegung signifikant [12]. Anstatt der Mehrgrößenregelung eines vermaschten Systems (vgl. Abschnitt 5.3) können die quasi modal entkoppelten Fahrdynamikgrößen v_{fx} , v_{fy} und ψ_f direkt geregelt werden.

5 Modellbildung

Eine wesentliche Grundlage für die Auslegung und Entwicklung eines leistungsfähigen Regelungssystems ist ein hinreichend genaues Modell der Regelstrecke. Für die Antriebsregelung des FTF, bei der für einen vorgegebenen Geschwindigkeits-Vektor die Ansteuerung der einzelnen Motoren berechnet werden müssen, sind Kenntnisse über die kinematischen und dynamischen Zusammenhänge des FTF notwendig (mechanischer Teil). Zudem wird ein hinreichend genaues Modell der elektrischen Antriebe benötigt (elektrischer Teil).

5.1 Mechanischer Teil

Das kinematische Verhalten ergibt sich aus der in Abbildung 4 dargestellten Antriebstopologie. Nach [7] ergibt sich die kinematische Beziehung zwischen der Bewegung des Fahrzeugschwerpunktes und den Drehzahlen der einzelnen Räder, welche Schlupffrei auf dem Untergrund abrollen, unter Einsatz der entsprechenden Parameter des FTF nach Gl. (2). Die Kinematikmatrix \underline{K} wird in Abhängigkeit von Abmessungen und Lage der Mecanum-Räder an dem Fahrzeugaufbau ermittelt. ω_j repräsentiert die jeweilige Drehzahl.

$$[v_{fx} \quad v_{fy} \quad \psi_f]^T = \underline{K} \cdot [\omega_1 \quad \omega_2 \quad \omega_3 \quad \omega_4]^T \quad (2)$$

Die Dynamik des FTF wird in Anlehnung an [13] mit Hilfe des Lagrange-Formalismus zweiter Art nach Gl. (3) hergeleitet. Dabei entspricht q_j den Freiheitsgraden des FTF, also den einzelnen Drehwinkeln der Räder, und Q_j den nicht konservativen Kräften, in diesem Fall den Momenten der Antriebsmotoren. Für den gesonderten Fall des FTF, welches sich lediglich auf ebenem Untergrund bewegt, wird die potentielle Energie vernachlässigt. Die gesamte Energie des Systems besteht somit lediglich aus der kinetischen Energie T , welche sich aus dem Geschwindigkeitsvektor des Fahrzeugaufbaus \dot{x}_f sowie den Raddrehzahlen $\underline{\omega}$ ergibt. Zur Vereinfachung wird der Massenschwerpunkt in der Mitte des Fahrzeugs angenommen und die Auflagekräfte an

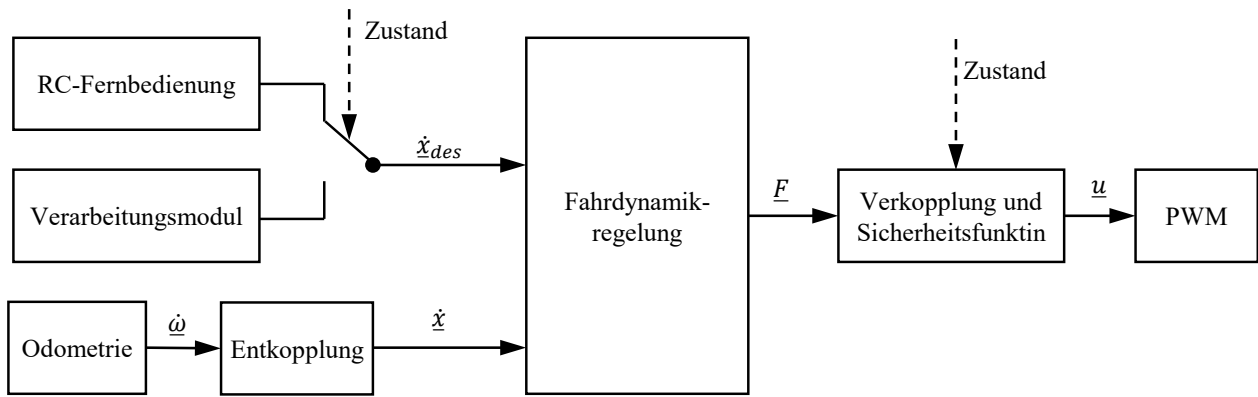


Abbildung 5: Vereinfachte Struktur der Funktion Antriebsregelung.

dem einzelnen Rad als Normalkraft beschrieben.

$$\frac{d}{dt} \left(\frac{\partial T}{\partial \dot{q}_j} \right) - \frac{\partial T}{\partial q_j} = Q_j \quad (3)$$

Somit ergibt sich die Dynamik des FTF nach Gl. (3). Unter Verwendung der kinematischen Beziehung nach Gl. (2) werden die Raddrehzahlen in die Bewegung des Fahrzeugschwerpunktes $\dot{x}_f = [v_{fx} \ v_{fy} \ \dot{\psi}_f]^T$ transformiert.

5.2 Elektrischer Teil

Jedes Mecanum-Rad (vgl. Abschnitt 4.1) wird von einem Gleichstrommotor (GSM) mit Getriebe angesteuert. Das erzeugte Moment Q_i ist Eingang für den mechanischen Teil des Modells. Abbildung 6 illustriert das physikalische Modell des GSM als elektrisches Ersatzschaltbild. Eingang ist die Klemmspannung u_{kl} .

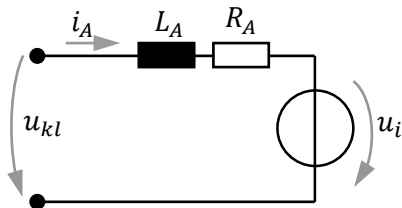


Abbildung 6: Elektrisches Ersatzschaltbild des GSM.

Durch Auswertung des Maschensatzes mit den Spannungsabfällen an der Induktivität L_A und dem Widerstand R_A sowie der induzierten Spannung u_i ergibt

sich das mathematische Modell für den Ankerstrom i_A :

$$u_{kl} = L_A \frac{di}{dt} + R_A \cdot i + u_i \quad (4)$$

Der Zusammenhang zwischen Winkelgeschwindigkeit des Mecanumrads j und der induzierten Spannung sowie dem Moment am Rad Q_j und dem Ankerstrom wird durch folgende Kopplungsgleichungen mit Hilfe der elektrischen und mechanischen Maschinenkonstante (c_1, c_2) beschrieben:

$$u_i = c_1 \cdot \omega_j \quad (5)$$

$$Q_j = c_2 \cdot i_A \quad (6)$$

5.3 Identifikation und Verifikation

Mit Hilfe von Messungen am realen System wird das Modell identifiziert und verifiziert. Zunächst erfolgen zur Parameteridentifikation des mechanischen und elektrischen Teilsystems Messungen im Frequenzbereich. Anschließend werden nicht-linearitäten und weitere im Frequenzbereich nicht darstellbare Zusammenhänge im Zeitbereich verifiziert.

Für die Identifikation des mechanischen Systems wird die Übertragungsmatrix nach Gl. (7) gebildet:

$$\underline{\underline{G}}(s) = \frac{\underline{\underline{\Omega}}(s)}{\underline{\underline{Q}}(s)} = \begin{bmatrix} G_{11} & G_{12} & G_{13} & G_{14} \\ G_{21} & G_{22} & G_{23} & G_{24} \\ G_{31} & G_{32} & G_{33} & G_{34} \\ G_{41} & G_{42} & G_{43} & G_{44} \end{bmatrix} \quad (7)$$

Die Abweichung zwischen Modell und Messung wird anschließend anhand der Modellparameter nach Gl. (8) minimiert. Aufgrund der starken Verkopplung

muss diese Minimierung für die gesamte Übertragungsmatrix stattfinden und kann nicht für einzelne Übertragungsfunktionen erfolgen.

$$\min \sum_i \left\| \underline{G}_{mess}(j\omega_i) - \underline{G}_{modell}(j\omega_i) \right\|^2 \quad (8)$$

Abbildung 7 illustriert exemplarisch das Ergebnis der Übertragungsfunktion G_{11} im Bode-Diagramm. Messung und Modell stimmen mit hoher Genauigkeit überein. Im niedrigen Frequenzbereich sind Abweichungen aufgrund des Drehzahlmessprinzips ersichtlich. Bei geringen Drehzahlen weisen die eingesetzten Inkrementalgeber eine geringe Auflösung auf. Im Frequenzbereich über 10^2 Hz nimmt der Einfluss der Totzeit der digitalen Informationsverarbeitung zu.

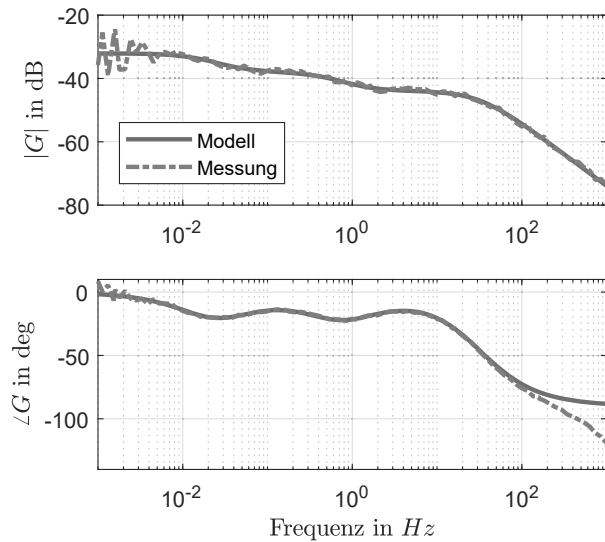


Abbildung 7: Bode-Diagramm der Übertragungsfunkt. G_{11} .

6 Funktionssynthese

Für die Funktionssynthese werden die Komponenten Entkopplung, Fahrdynamikregelung und Verkopplung nach Abbildung 5 modellbasiert ausgelegt. Anschließend erfolgen Erprobung und Optimierung mittels MiL-Simulationen anhand des in Kapitel 5 erstellten Modells.

6.1 Entkopplung und Verkopplung

Zur Entkopplung wird ein Zusammenhang $f(\underline{\omega})$ gesucht, welcher anhand der gemessenen Winkelgeschwin-

digkeiten der Mecanum-Räder auf den quasi modalen Geschwindigkeitsvektor $\dot{\underline{x}}$ des FTF schließt:

$$\dot{\underline{x}} = f(\underline{\omega}) \quad (9)$$

Dieser Zusammenhang ist durch die ermittelte Kinematik-Matrix nach Gl. (2) gegeben.

Für die Verkopplung wird ein Zusammenhang $g(\underline{F})$ gesucht, welcher anhand des Antriebskraftvektors die Ansteuerspannung der Elektromotoren \underline{u} berechnet:

$$\underline{u} = g(\underline{F}) \quad (10)$$

Zur Bestimmung von $g(\underline{F})$ wird die inverse Kinematik des FTF benötigt. Aufgrund der Dimensionen, ist eine Invertierung der Kinematik-Matrix \underline{K}^{-1} jedoch nicht möglich. Daher wird die Moore-Penrose-Inverse \underline{K}^+ nach [14] gebildet. Unter Verwendung des elektrischen Modellteils ergibt sich für die stationäre Verkopplung:

$$\underline{u} = \frac{R_A}{c_2} \cdot \underline{K}^+ \cdot \underline{F} + c_1 \cdot \underline{\omega} \quad (11)$$

6.2 Fahrdynamikregelung

Durch die forcierte Struktur mit Entkopplung und Verkopplung lassen sich die drei modalen Bewegungsgrößen der Ebene getrennt voneinander regeln. Eingesetzt werden PI-Regler, welche mit dem Betragsoptimum im Frequenzbereich ausgelegt und anschließend mittels MiL-Simulationen hinsichtlich der in Kapitel 4 definierten Anforderungen optimiert werden. Abbildung 8 stellt exemplarisch die Simulationsergebnisse bei sprunghörmiger Anregung mit 1 m/s in x-Richtung dar. Im obersten Diagramm ist das gute Führungsverhalten von v_{fx} ersichtlich. In den beiden unteren Diagrammen wird deutlich, dass die Anregung in x-Richtung keinen Einfluss auf v_{fy} und $\dot{\psi}_f$ hat. Die Ergebnisse sind repräsentativ für sämtliche Anregungen, wonach die Freiheitsgrade unabhängig voneinander geregelt werden können.

7 Realisierung und Verifikation

Nach der Funktionsauslegung und -optimierung durch diverse MiL-Simulationen, erfolgt mittels automatischem Codegenerator der LoRra-Plattform die Transformation in C-Code. Dieser wird wiederum offline durch SiL-Simulationen getestet und optimiert. Bei hinreichenden Ergebnissen erfolgt über ein Real-Time-

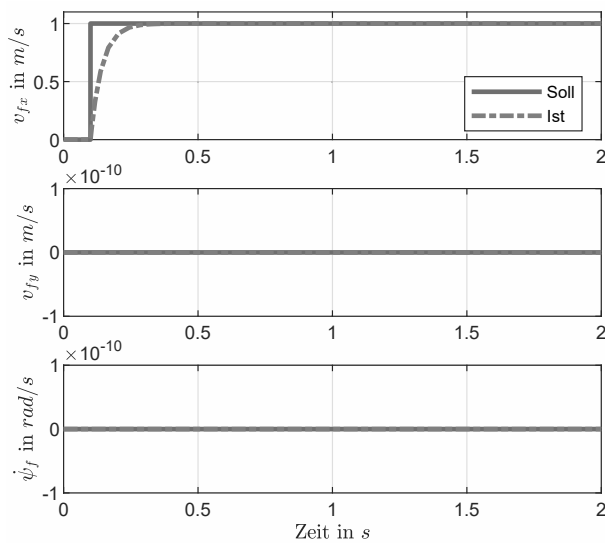


Abbildung 8: Sprungantwort des geregelten Systems bei Anregung in x-Richtung.

Interface die Verknüpfung der Funktionsschnittstellen mit den Schnittstellen der Low-Cost-Echtzeithardware. Abbildung 9 illustriert den resultierenden Prototypen des FTF. Dieser wird eingesetzt, um sowohl die Funktionen des Antriebsmoduls, als auch die entwickelten intelligenten autonomen Fahrfunktionen im Labortestfeld unter Echtzeitbedingungen zu erproben und zu optimieren.



Abbildung 9: Prototyp des FTF.

8 Zusammenfassung und Ausblick

Der vorliegende Beitrag behandelt die modellbasierte Entwicklung des Antriebsmoduls für ein vernetztes FTF als Mobilitätsplattform in einem cyber-physischen Labortestfeld. Unter Anwendung einer ganzheitlichen, durchgängigen Methode und unter Einsatz der Low-Cost RCP-Entwicklungsplattform LoRra werden die wesentlichen Modelle und Entwicklungsschritte vorgestellt.

Am Stand des Wissens wird das Einsatzziel der FTF als Mobilitätsplattform verdeutlicht. Grundlage sind vorhandene Funktionen und ein hohes Maß an Erfahrung durch den Einsatz in Industrieanlagen. Das Konzept des modular aufgebauten FTF sieht die vier wesentlichen Bestandteile Kommunikationsmodul, Perzeptionsmodul, Verarbeitungsmodul sowie Antriebs- und Energiemodul vor. Dieser Beitrag konzentrierte sich dabei auf das Antriebsmodul. Über Modularisierung und Hierarchisierung wurden definierte Schnittstellen geschaffen und sowohl das mechanische als auch das informationstechnische Konzept des Antriebsmoduls dargelegt. Ein identifiziertes, verifiziertes Modell bildete die Grundlage für die Funktionssynthese. Die Funktion Antriebsregelung zur Erfüllung der Teilaufgabe *antreiben* wurde modellbasiert in der Low-Cost-Entwicklungsplattform LoRra ausgelegt sowie mittels MiL-Simulationen optimiert und verifiziert. Im nächsten Schritt wurde über die automatische Codegenerierung ausführbarer C-Code generiert und über ein Real-Time-Interface auf einen Prototypen zur weiteren Optimierung und Absicherung unter Echtzeitbedingungen implementiert.

Weitere Arbeiten befassen sich mit der Integration der einzelnen Module zum autonom fahrenden FTF sowie deren Echtzeitoptimierung und -test. Diese bilden die Grundlage für weitere Forschungen zur intelligenten Mobilität in cyber-physischen Verkehrssystemen. Durch Einsatz im cyber-physischen Labortestfeld für intelligente Mobilitätsanwendungen wird im Rahmen des Niedersächsischen Zukunftslabors Mobilität mit zahlreichen Forschungs- und Anwendungspartnern an zukünftigen innovativen Plattformen, Komponenten und Funktionen geforscht.

Danksagung

Gefördert vom Niedersächsischen Ministerium für Wissenschaft und Kultur unter Fördernummer ZN3495 im Niedersächsischen Vorab der VolkswagenStiftung und betreut vom Zentrum für digitale Innovationen (ZDIN).



Referenzen

- [1] Kim H, Song M, Doerzaph Z. Real-World Use of Partially Automated Driving Systems and Driver Impressions. *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*. 2020; 64(1):1092–1093. DOI: 10.1177/1071181320641262.
- [2] Bundesministerium für Verkehr und digitale Infrastruktur, ed. *Bericht zum Stand der Umsetzung der Strategie automatisiertes und vernetztes Fahren*. 2017. URL <https://www.bmvi.de/SharedDocs/DE/Publikationen/DG/bericht-avf.pdf>
- [3] Liu-Henke X, Jacobitz S, Gollner M, Zhang J, Scherler S, Yarom OA. Cyber-physical Industry 4.0 laboratory test field to simulate self-optimizing intralogistics. In: *2020 19th International Conference on Mechatronics - Mechatronika (ME)*. Prague, Czech Republic. 2020 Dec; pp. 1–6. DOI: 10.1109/ME49197.2020.9286614.
- [4] Ullrich G, Albrecht T. *Fahrerlose Transportsysteme: Eine Fibel - mit Praxisanwendungen - zur Technik - für die Planung*. Wiesbaden: Springer Fachmedien, 3rd ed. 2019. DOI: 10.1007/978-3-658-27472-6.
- [5] Schwarz C. Untersuchung zur Steigerbarkeit von Flexibilität, Performanz und Erweiterbarkeit von Fahrerlosen Transportsystemen durch den Einsatz dezentraler Steuerungstechniken. Dissertation, Universität Oldenburg. 2014.
- [6] VDI-Gesellschaft Produktion und Logistik, ed. *Kompatibilität von fahrerlosen Transportsystemen (FTS)*. VDI 4451. 1998 Feb.
- [7] Liu-Henke X, Scherler S, Göllner M, Maisik J, Fritsch M. Simulationsgestützte Konzeption der Antriebstopologie eines fahrerlosen Transportfahrzeugs. In: *Workshop der ASIM/GI-Fachgruppen*. Heilbronn, Germany. 2018 Mar; pp. 169–174.
- [8] Liu-Henke X, Duym S. Modellgestützte Funktionsabsicherung des vernetzten mechatronischen Kraftfahrzeugs. In: *Mechatronik 2005 - Innovative Produktentwicklung*, vol. 1892 of *VDI-Berichte*. Düsseldorf: VDI-Verl. 2005; pp. 1073–1090.
- [9] Jacobitz S, Liu-Henke X. The Seamless Low-cost Development Platform LoRra for Model based Systems Engineering. In: *Proceedings of the 8th International Conference on Model-Driven Engineering and Software Development*. SCITEPRESS - Science and Technology Publications. 2020 Feb; pp. 57–64. DOI: 10.5220/0008993500570064.
- [10] Göllner M, Liu-Henke X, Frerichs L. Analyse und Simulation des Kraftübertragungsverhaltens von Mecanum-Rädern. In: *25. Symposium Simulationstechnik - ASIM 2020*. virtuelle Tagung. 2020 Oct; pp. 89–98. DOI: 10.11128/arep.59.
- [11] Li Y, Dai S, Zhao L, Yan X, Shi Y. Topological Design Methods for Mecanum Wheel Configurations of an Omnidirectional Mobile Robot. *Symmetry*. 2019; 11(10):1268. DOI: 10.3390/sym11101268.
- [12] Liu-Henke X. *Mechatronische Entwicklung der aktiven Feder-Neigetechnik für das Schienenfahrzeug RailCab*, vol. 589 of *Fortschritt-Berichte VDI Reihe 12, Verkehrstechnik/Fahrzeugtechnik*. Düsseldorf / Paderborn: VDI-Verlag. 2005.
- [13] Zhang J, Liu-Henke X. Model-based design of the vehicle dynamics control for an omnidirectional automated guided vehicle (AGV). In: *2020 International Conference Mechatronic Systems and Materials (MSM)*. 01.07.2020 - 03.07.2020; pp. 1–6. DOI: 10.1109/MSM49833.2020.9202248.
- [14] Stoer J, Bauer FL. *Numerische Mathematik: Eine Einführung; unter Berücksichtigung von Vorlesungen von F. L. Bauer*. Springer-Lehrbuch. Berlin: Springer, 8th ed. 2002.

Simulation eines mobilen Manipulators mit Gazebo

Kai Mario Kriegel^{1*}, Steffen Ring², Christian Klöppelt³, Dagmar Meyer¹,

¹Fakultät Elektrotechnik, Ostfalia Hochschule für angewandte Wissenschaften, Salzdahlumer Str. 46-48, 38302 Wolfenbüttel; *k.kriegel@ostfalia.de

²Fakultät Fahrzeugtechnik, Ostfalia Hochschule für angewandte Wissenschaften, Robert-Koch-Platz 8A, 38440 Wolfsburg

³Institut für Regelungstechnik, Leibniz Universität Hannover, Appelstraße 11/11a, 30167 Hannover

Kurzfassung. In diesem Dokument wird beschrieben, wie ein mobiler Manipulator, der in einem Pflegeumfeld eingesetzt werden soll, in einer Simulationsumgebung nachgebildet wird. Die wesentlichen Bestandteile des Roboters sind eine autonome mobile Plattform, ein kollaborativer Leichtbauroboterarm und ein Drei-Finger-Greifer. Die Kommunikation zwischen den Komponenten und die Softwarefunktionalitäten basieren auf dem Robot Operating System (ROS). Als Simulationsumgebung wird Gazebo verwendet. Die Simulation ermöglicht die Erprobung von Operationen und den Test von Funktionen unter gleichbleibenden Bedingungen, bevor diese am realen Robotersystem durchgeführt werden. Dies führt zu einem geringeren Zeitaufwand und ermöglicht das simultane Entwickeln von Funktionen.

Einleitung

Im Forschungsvorhaben PersonA-PP (Persönliche Assistenz für Patienten in der Pflege) wird ein Roboterassistent mit innovativer Benutzungsschnittstelle entwickelt, der in Wohn- und Pflegeeinrichtungen lebenden Menschen als persönlicher Assistent zur Verfügung steht. Das robotische System soll dabei für mehrere Personen zuständig sein und diese durch einfache Handreichungen wie das Holen und Zureichen von Gegenständen unterstützen, um dem Pflegepersonal mehr Zeit für patientenbezogene Aufgaben und menschliche Zuwendung einzuräumen [1]. Eine weitere Aufgabe des Roboters besteht in der Unterstützung des Personals vor und während der von den Bewohnerinnen und Bewohnern im Speisesaal eingenommenen Mahlzeiten, z. B. durch das Verteilen von Kaffee- und Teekannen auf den Tischen.

Für das aus einer mobilen Plattform und einem Roboterarm bestehende autonome System ist die Entwicklung zahlreicher Basisfunktionen wie z. B. Umfeldwahrnehmung, Selbstlokalisierung, Navigation in einer dynamischen Umgebung, Erkennung und Lokalisation von Objekten usw. erforderlich, um die zur Bewältigung der Aufgaben notwendigen komplexen Abläufe implementieren zu können. Test und systematischer Vergleich unterschiedlicher Lösungsansätze gestalten sich auf dem Zielsystem relativ aufwändig. Aufgrund der Interaktion des Systems mit seiner Umwelt ist darüber hinaus die Berücksichtigung stark variierender Szenarien (individuell gestaltete Pflegezimmer, Flure, Speisesäle) notwendig.

Um die Entwicklung der komplexen Systemsoftware zu vereinfachen und zu beschleunigen, wird eine Software-in-the Loop (SiL) Entwicklungs- und Testumgebung benötigt, die den direkten Einsatz der erfolgreich getesteten Software auf dem Zielsystem ohne Modifikationen ermöglicht und mit deren Hilfe die vorhandene Sensorik sowie unterschiedliche Umgebungen simuliert werden können [2, 3].

Das vorliegende Dokument beschreibt die Modellierung des auf dem Robot Operating System (ROS) basierenden Systems, seiner Hardwareschnittstellen und seines dynamischen Verhaltens mithilfe der Simulationsumgebung Gazebo.

Ausgehend vom im nachfolgenden Kapitel skizzierten Stand der Technik werden in Kapitel 2 zunächst die Software-Frameworks ROS und Gazebo eingeführt. Die Kapitel 3 und 4 beschreiben die Modellierung der einzelnen Systemkomponenten sowie die Nachbildung der Umgebung. Eine Zusammenfassung der bisher erreichten Ziele und ein Ausblick auf weitere Arbeiten (Kapitel 5) schließen die Ausführungen ab.

1 Stand der Technik

Für die Simulation von robotischen Systemen mit auf dem Framework ROS basierender Systemsoftware werden die Simulationsumgebungen Gazebo [4] und CoppeliaSim [5] (vormals unter der Bezeichnung V-REP bekannt [6]) vielfach erfolgreich eingesetzt. Bei CoppeliaSim handelt es sich um ein kommerzielles Produkt, Gazebo ist hingegen ein Open-Source Simulator.

Gazebo kommt beispielsweise für die Simulation mobiler Roboter zum Einsatz. In [7] wird die Simulation einer einfachen mobilen Plattform (TurtleBot) in Gazebo mit dem primären Ziel des Tests von SLAM-Algorithmen beschrieben. Die Autoren von [8] beschreiben ein kinematisches und dynamisches Modell für die Kobuki-Mobilbasis des TurtleBot2. Sie nutzen Gazebo für den Vergleich des Modells mit dem realen Systemverhalten. In [9] wird der Ansatz verfolgt, durch den Einsatz von Gazebo und ROS die in der Simulationsumgebung für mobile Roboter mit Differenzialantrieb entwickelte Software direkt auf dem Zielsystem einzusetzen. Die Autoren beschreiben die Vorgehensweise zur Modellierung mobiler Roboter sowie zur Definition von Umgebungen in Gazebo. Die Modellierung eines intelligenten Rollstuhls sowie eine Systemarchitektur zum gleichzeitigen Testen des simulierten und des realen Rollstuhls auf der Basis von ROS werden in [10] vorgestellt.

Auch für die Simulation von stationären Roboterarmen wird Gazebo verwendet. In [11] wird beispielsweise ein Modell für einen Roboterarm mit 6 Freiheitsgraden in Gazebo erstellt, wobei die Schätzung der Trägheitsmomente der einzelnen Elemente des Armes mithilfe des Steinerschen Satzes erfolgt. Primäres Ziel der Simulation ist hier die Unterstützung der Auslegung der Gelenkregelungen. Die Erstellung eines Simulationsmodells für ein System mit einem flexiblen Greifer (Soft Gripper) wird in [12] beschrieben.

Eine Anwendung der Simulation mit Gazebo für ein komplexes System mit einer Vielzahl von Sensoren ist in [13] zu finden. In dem Artikel wird der vom Fraunhofer IPA entwickelte Serviceroboter Care-O-bot3 beschrieben, der inzwischen durch das Nachfolgemodell Care-O-bot4 abgelöst wurde. [14] befasst sich mit der Modellierung und Simulation des humanoiden Roboters Aldebaran NAO.

Die Zielstellung der im vorliegenden Dokument beschriebenen Simulation vereint zahlreiche Teilaspekte, die in den genannten Veröffentlichungen adressiert werden. Ein heterogenes System aus

Teilkomponenten unterschiedlicher Hersteller mit diversen Sensoren und Aktoren soll unter Berücksichtigung seines dynamischen Verhaltens so nachgebildet werden, dass die auf der Basis dieser Simulation entwickelte Systemsoftware ohne Modifikationen auch auf dem realen Zielsystem eingesetzt werden kann. Darüber hinaus soll die Erprobung der Interaktion mit einer frei konfigurierbaren Umgebung ermöglicht werden.

2 ROS und Gazebo

Das bestehende Robotersystem nutzt in weiten Teilen ROS, das im Abschnitt 2.1 kurz eingeführt wird. Es existieren diverse mit ROS kompatible Simulationsumgebungen, u. a. Gazebo, CoppeliaSim (früher V-REP), ARGoS, MORSE und Unity die in [15, 16, 17, 18] verglichen werden. Gründe für die Wahl des in Abschnitt 2.2 vorgestellten Gazebo-Simulators für das Projekt sind die Verfügbarkeit von Gazebo-Implementierungen für die meisten der eingesetzten Komponenten, der verfolgte Ansatz keine proprietäre Software einzusetzen sowie die hohe Kompatibilität von ROS und Gazebo.

2.1 ROS

ROS ist ein Open Source Framework, das für die Entwicklung und den Betrieb von Robotersystemen genutzt werden kann. ROS bietet neben einer Kommunikationsschicht für die Interprozesskommunikation zahlreiche Bibliotheken zur Einbindung von Sensoren und Aktoren. Viele in der Robotik verbreitete Algorithmen sind bereits in ROS verfügbar und können mit geringem Aufwand eingebunden werden. [19] Im folgenden wird kurz die Nomenklatur beschrieben (vgl. Abbildung 1).

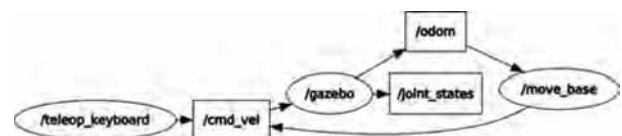


Abbildung 1: Ausschnitt des ROS-Netzwerkes mit integrierter Gazebo-Simulation. Ellipsen symbolisieren Nodes, Rechtecke Topics. Zwei Nodes können Geschwindigkeitsvektoren an Gazebo senden, das die aus der Simulation resultierenden Odometrie-Informationen und Gelenkpositionen veröffentlicht.

Package ROS besteht aus zahlreichen Packages, wobei jedes Package eine abgeschlossene Funktionalität abbildet. Das System kann durch eigene Packages beliebig erweitert werden.

Node Ein Node ist ein Prozess, der die ROS-API nutzt, wobei der Begriff Node mit dem Begriff Software-Modul gleichbedeutend ist [19]. Ein auf ROS basierendes System umfasst in der Regel mehrere Nodes, die miteinander kommunizieren. Jeder Node kann mit dem ihm zugewiesenen, arbiträren aber eindeutigen Namen identifiziert werden.

Message Zwischen den Nodes können Messages ausgetauscht werden. Der Aufbau einer Nachricht wird in einer Interface Definition Language (IDL) beschrieben. Mit dieser Beschreibung können ROS-Nodes die in unterschiedlichen Programmiersprachen realisiert sein können miteinander kommunizieren.

Topic Nachrichten können asynchron über ein Topic zwischen zwei Nodes unidirektional ausgetauscht werden [20]. Ein Topic ist die Bezeichnung für einen Bus, der von ROS genutzt wird, um die Kommunikation zwischen Nodes nach dem Publish/Subscribe-Prinzip zu ermöglichen [19, 21]. Um Nachrichten eines Topics zu erhalten, müssen sich die Nodes unter diesem Topic einschreiben. Ebenfalls können sie Nachrichten unter einem Topic veröffentlichen, indem sie als Publisher auftreten.

Service Neben der asynchronen kann eine synchrone, bidirektionale Kommunikation über Services erfolgen. Dabei wartet der anfragende Client-Node auf die Antwort des anbietenden Nodes, wobei der Aufbau von Anfrage und Antwort ebenfalls mithilfe einer IDL beschrieben wird.

2.2 Gazebo

Gazebo ist eine Open Source Robotersimulationsumgebung, die es ermöglicht das Verhalten eines oder mehrerer Roboter in einer 3D-Umgebung zu simulieren. Neben der Simulation von Texturen, Schatten und Licht können auch Aktoren, Sensoren und deren Rauschen in Gazebo über Plugins simuliert werden. Die Funktionalität ist durch eigene Plugins erweiterbar. Die Kopplung mit ROS erfolgt über das Package `gazebo_ros_pkgs` [22], wodurch die Hardware

eines bestehenden Systems in Gazebo abgebildet werden kann und keine oder nur geringe Änderungen an der übrigen Systemsoftware nötig sind. Gazebo unterstützt verschiedene Physics-Engines und bietet eine grafische Oberfläche[4, 23]. In der Abbildung 1 ist ein Ausschnitt des ROS-Netzwerks dargestellt, in das Gazebo eingebunden ist.

3 Simulationsmodell des Roboters



Abbildung 2: Simulationsmodell des Roboters. 1. Stereokamera, 2. 3-Finger-Greifer, 3. Roboterarm, 4. UWB-Anker, 5. Mobile Plattform.

In diesem Kapitel werden die Hardwarekomponenten des in Abbildung 2 gezeigten Robotersystems erläutert, die mit ihren Schnittstellen in der Simulation nachgebildet werden. Der Roboter besteht im wesentlichen aus einer mobilen Plattform MP400 von Neobotix, einem kollaborierenden Roboter Panda von Franka Emika und einem 3-Finger-Greifer von RobotIQ. Als zusätzliche Sensorik werden eine Intel Realsense D435i Stereokamera und ein Ultrabreitband(UWB)-Anker verwendet.

3.1 Mobile Plattform

Die mobile Plattform wird über zwei Räder angetrieben, mit denen ein differenzieller Antrieb realisiert ist. Über einen Sick S300 Expert Laserscanner und ein Odometer, dass die Drehraten der Räder nutzt, kann die Plattform über weitere ROS-Nodes

lokalisiert werden. Der Laserscanner überwacht zusätzlich ein geschwindigkeitsabhängiges Schutzfeld, wodurch Kollisionen mit der Umgebung durch Abschaltung der Motoren vermieden werden. Da dies kein regulärer Anwendungsfall ist, wird dies in der Simulation nicht berücksichtigt. Das dynamische Verhalten der Plattform und die ROS-Schnittstellen der Sensorik und Aktorik werden in der Simulation nachgebildet. Als Grundmodell für die Simulation der mobilen Plattform dient ein unter [24] vom Hersteller Neobotix bereitgestelltes, auf die eingesetzte mobile Plattform angepasstes Package. Plugins zur Implementierung eines Laserscanners und eines differenziellen Antriebs sind in Gazebo verfügbar und müssen nur entsprechend parametrisiert werden. Abbildung 3 zeigt die berechnete Position aus den Odometriedaten, nachdem der reale und der simulierte Roboter mit einem zeitlichen Verlauf von Geschwindigkeitsvektoren beaufschlagt wurden. Die Abweichung in der X-Y-Position resultiert hauptsächlich aus einem Winkeloffset, der beim Anfahren entsteht. Da die Geschwindigkeiten durch einen überlagerten Pfadplaner vorgegeben werden, ist diese geringe Abweichung für unseren Anwendungsfall zu vernachlässigen.

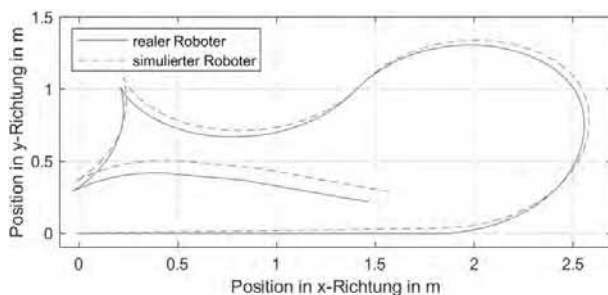


Abbildung 3: Beispielhafter Fahrweg des Roboters

3.2 Kollaborierender Roboterarm

Der kollaborierende Roboterarm Franka Emika Panda hat sieben Freiheitsgrade. Die Geometrie der einzelnen Körper ist unter [25] vom Hersteller frei zugänglich. Die Simulation des dynamischen Verhaltens des Roboters geschieht auf Basis der Bewegungsgleichungen, welche den Zusammenhang zwischen den Gelenkdrehmomenten sowie den Gelenkpositionen, -geschwindigkeiten und -beschleunigungen beschreiben. Aus der Geometrie der einzelnen Armglieder lassen sich die symbolischen Bewegungs-

gleichungen des Roboters, beispielsweise über den Lagrange-Formalismus, bestimmen [26, Kapitel 7]. Die Parameter dieses dynamischen Modells stellen die Massen, Trägheitstensoren sowie Positionen der Schwerpunkte der einzelnen Armglieder dar. Da diese physikalischen Parameter unbekannt sind, müssen sie, um eine präzise Simulation des Roboterarmes zu ermöglichen, im Rahmen einer Modellidentifikation [27] bestimmt werden.

Hierzu werden zunächst die physikalischen in die identifizierbaren Parameter, die sog. Basisparameter, umgerechnet [28, 29]. Anschließend werden die einzelnen Gelenke mit zur Identifikation geeigneten Drehmomentsignalen angeregt und die Gelenkpositionen, -geschwindigkeiten und -beschleunigungen gemessen und gefiltert [30, 31, 32]. Im Anschluss können diese Messdaten genutzt werden, um die Parameter des Roboters zu schätzen. Da das dynamische Modell affin in den Basisparametern ist, kann dies über einfache lineare Regressionsverfahren geschehen [33, Kapitel 12].

Die zur Simulation des Roboters verwendete Open Dynamics Engine basiert auf den physikalischen Parametern der Armglieder, weshalb diese im Anschluss an die Modellidentifikation aus den Basisparametern bestimmt werden müssen. Dies kann durch Lösen eines semidefiniten Programms [34], dessen Nebenbedingungen den Lösungsraum auf physikalisch zulässige Parameter [35, 36, 37, 38] beschränkt, geschehen. Da sich einige der physikalischen Parameter nicht oder nur teilweise identifizieren lassen, entsprechen die auf diese Weise bestimmten Massen, Trägheitstensoren und Positionen der Schwerpunkte der Armglieder nicht den realen Werten. Jedoch sind sie physikalisch zulässig und ggf. auch sinnvoll und eignen sich daher gut zur Simulation der Roboterdynamik. Vergleicht man das Verhalten des Roboters beim Beaufschlagen von Trajektorien kann gezeigt werden, dass nur geringe Abweichungen zwischen dem simulierten und realen Roboter auftreten. Das kollaborierende Verhalten des Roboters, das zum Anhalten im Fall einer Kollision führt, wird jedoch nicht berücksichtigt.

3.3 3-Finger-Greifer

Der 3-Finger-Greifer von RobotIQ bietet vier Greifmodi. Jeder einzelne Finger besteht aus drei Gliedern. Diese werden untereinander mit jeweils zwei Gelenkstangen unterstützt. Diese ermöglichen sowohl ein Versteifen des Fingers zum Formen von Griffmustern

als auch ein Einknicken zum Umgreifen eines Gegenstandes. Jeder Finger kann mit einem Motor in Richtung Handballen angewinkelt werden. Zusätzlich können die beiden Finger der einen Seite mit zwei weiteren Motoren zusammen- und auseinandergefahren werden. Der unteraktuierte Greifer passt sich durch die mechanische Konstruktion adaptiv an das gegriffene Objekt an. Für die Abbildung des Greifers in der Simulation stehen zwei verschiedene Modelle zur Verfügung [39]. Ein Modell bietet eine einfache Abbildung und beschränkt sich dabei auf die essentiell wichtigen Elemente der Hand und verzichtet gänzlich auf die Gelenkstangen. Für dieses Modell muss das adaptive Verhalten durch Vorgabe der jeweiligen Gelenkposition angegeben werden. Dies ist durch die real existierende Mechanik allerdings nicht ohne weiteres möglich. Das zweite Modell bildet diese mit ab. Es wird in der hier beschriebenen Umsetzung verwendet, da für dieses ein Gazebo-Plugin verfügbar ist, welches die Steuerung des simulierten Greifers ermöglicht. Mit diesem Modell ist das Greifen von einfachen Gegenständen möglich. Das adaptive Greifen des realen Greifers wird dabei jedoch nicht korrekt abgebildet. Der simulierte Greifer beschränkt sich auf das Zusammen- und Auseinanderfahren der Finger. Dennoch können die Bewegung und die Positionierung des Greifers simuliert werden. Dies erlaubt das wiederholte Erproben und Trainieren von Abläufen, jedoch nicht von Greifstrategien.

3.4 Stereokamera

Am Endeffektor des Roboters ist eine Intel Realsense D435i Stereokamera montiert. Die Kamera, die neben Farb- auch Tiefeninformationen liefert, wird verwendet um Objekte zu erkennen, deren Position zu bestimmen sowie Kollisionen mit der Umgebung beim Bewegen des Roboterarmes zu vermeiden. Für die Simulation einer Tiefenkamera bietet Gazebo die Möglichkeit ein Kamera-Plugin einzubinden. Das Plugin generiert die Tiefen- und Farbinformationen aus der Simulation und veröffentlicht diese an das ROS-Netzwerk. Das visuelle Modell der Kamera ist unter [40] verfügbar. Die Implementierung in Gazebo basiert auf [41]. Die im PointCloud2-Format übertragenen Tiefeninformationen [42] sind mit denen des realen Robotersystems vergleichbar. Die Farbbilder sind nicht sehr realitätsnah, weshalb die Objektdetektion speziell auf die Daten der Simulation angepasst werden muss. Die von Gazebo gelieferten Sensorinformationen der Kamera sind in der Abbildung 5 unten zu sehen.

3.5 UWB-Anker

Für die Lokalisierung von Objekten wird ein System verwendet, das auf der Ultrabreitband-Technologie basiert. Wie in Abbildung 2 dargestellt, ist an der mobilen Plattform ein UWB-Anker befestigt. Dieser kommuniziert mit den im Raum verteilten UWB-Tags, die an Objekten angebracht werden, mit denen der Roboter interagieren soll. Anhand der Signallaufzeit ist die Bestimmung der Distanz zwischen Anker und Tag möglich. Mit dieser Information soll das Robotersystem Objekte lokalisieren, wobei eine präzise Lokalisation aufgrund der begrenzten Genauigkeit durch Mehrwegeausbreitung nicht möglich ist. Die exakte Bestimmung der Position des zu findenden Gegenstands übernimmt die Stereokamera. Das UWB-System wurde in die Simulationsumgebung integriert. So ist es möglich beliebige Gegenstände mit konfigurierbarem physikalischen Verhalten und optischer Gestaltung zu erzeugen. Durch Verwendung der Gazebo-API [43] wird zyklisch die Position der erzeugten Objekte abgefragt und der Abstand zu diesen berechnet. Diese werden zusammen mit dem Namen des jeweiligen Objektes an das Gesamtsystem veröffentlicht. Um das stochastische Verhalten durch Mehrwegeausbreitung und andere Einflüsse auf die Signallaufzeiten der UWB-Signale abzubilden, wird der berechnete Abstand mit einer Rayleigh-Verteilung angereichert [44, 45]. Die Integration des Distanzmesssystems in die Simulation erleichtert die Entwicklung und Erprobung von Verfahren, mit denen die Position der Objekte bestimmt werden kann. Beliebig viele virtuelle Tags können an variablen Positionen im Raum erstellt werden, ohne dass diese in der Realität zur Verfügung stehen müssen.

4 Nachbildung der Umgebung

Der Roboter wird zunächst in einem Laborraum (Nachbildung einer Pflegeumgebung) eingesetzt, in dem gewählte Anwendungsszenarien erprobt werden können. Dieser Laborraum wurde für die Validierung der Simulation näherungsweise nachgebildet. Dabei muss ein Kompromiss zwischen Detailtreue und abstrahierter Umgebung gefunden werden, da die Komplexität der Umgebung die für die Simulationsausführung benötigte Rechenzeit beeinflusst [46]. Zur Nachbildung einer Umgebung können unterschiedliche Model-Files eingebunden werden. Abbildung 4 zeigt den Roboter in der Simulationsumgebung. Die blauen Trennwände unterteilen den Raum in mehrere Sektoren. Durch

die Nachbildung des Laborraumes kann gezeigt werden, dass die Lokalisierung und Pfadplanung innerhalb der Simulation dem Verhalten des Roboters im realen Laborraum entspricht. In der Abbildung 5 sind ein Ausschnitt der Simulationsumgebung und die daraus generierten Sensorinformationen zu sehen, die an das ROS-Netzwerk veröffentlicht werden.

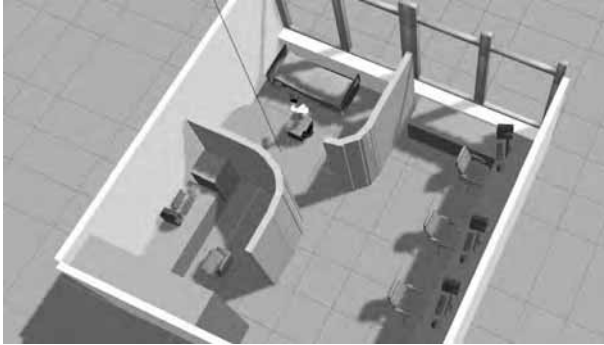


Abbildung 4: In Gazebo erstellte Nachbildung des Laborraumes mit dem Roboter in der Nähe des Pflegebetts zur Validierung der Simulation.

5 Fazit und Ausblick

Eine SiL-Umgebung wurde erfolgreich erstellt und anhand einer Nachbildung des Laborraumes validiert. Die Simulation ermöglicht die Entwicklung und den Test von Funktionalitäten für Teilbereiche und das Gesamtsystem. Sie umfasst die Bereitstellung von Distanzmessungen zwischen UWB-Anker und -Tags, das Anfahren von Positionen mit der mobilen Plattform und das Bewegen des Roboterarms und des Greifers. Das Kamerabild kann verwendet werden um Kollisionen beim Bewegen des Roboters zu vermeiden. Zur Verwendung der Objekterkennung aus den simulierten Kamerabildern muss diese jedoch speziell auf die Bilder der Simulation angepasst werden, da diese eine zu große Abweichung zu den Bildern einer echten Kamera haben.

Einige Teilaspekte des Systems werden in der Simulation nicht berücksichtigt. Die geschwindigkeitsabhängige Notabschaltung der mobilen Plattform sowie die Kollisionserkennung und automatische Abschaltung des Roboterarmes sind nicht implementiert. Es handelt sich zwar nicht um gewünschte Anwendungsfälle, aber für Untersuchungen mit beweglichen Akteuren müssen diese Funktionen hinzugefügt werden. Um Detailfunktionen mithilfe der Simulation zu unter-

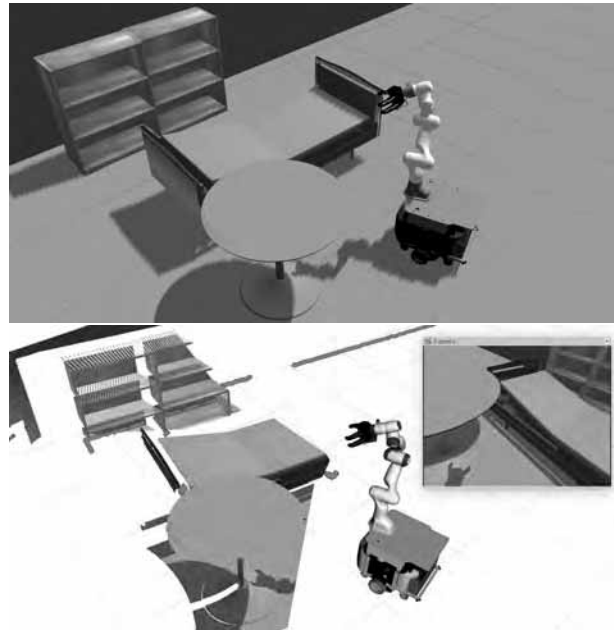


Abbildung 5: Simulationsumgebung (oben) und daraus gewonnene Sensorinformationen (unten). Zu sehen sind die Tiefeninformationen der Stereokamera (links), die Sensorinformationen des Laserscanners (rot) und das ebenfalls von der Kamera generierte Farbbild (rechts). Auch die Ausrichtung des Roboterarms und die Odometriedaten der mobilen Plattform werden von der Simulationsumgebung bereitgestellt.

suchen, müssen die jeweiligen Modelle verfeinert werden. Zum Beispiel ist das Testen von Greifarten innerhalb der Simulation durch die Näherung des adaptiven Verhaltens derzeit nicht realitätsnah möglich.

Zukünftig sollen weitere Szenarien in der Simulation erstellt werden, die es ermöglichen, das System in seinem späteren Einsatzfeld zu erproben. So ist es zum Beispiel möglich, einen Abschnitt aus einem realen Pflegeheim nachzubilden und damit die Praktikabilität zu untersuchen und zu verbessern, bevor tatsächliche Versuche in einer derartigen Umgebung stattfinden.

Danksagung

Dieser Beitrag wurde im Rahmen des Projekts Persönliche Assistenz für Patienten in der Pflege (PersonA-PP) durch den Europäischen Fonds für regionale Entwicklung (EFRE) und das Land Niedersachsen unter dem Förderkennzeichen ZW 6-85023423 gefördert. Die

Verantwortung für den Inhalt liegt bei den Autoren.



Referenzen

- [1] Ostfalia. Projektinformation PersonA-PP - Persönliche Assistenz für Patienten in der Pflege. 2018. Zugriff: 2021-02-17.
URL <https://www.ostfalia.de/cms/de/forschung/forschung-ostfalia/forschungsprojekte/research-info-detail/index.html?lang=de&nodeId=7d4c3267-a674-4a21-87c2-ff21f5cd44dd>
- [2] Jaensch F, Csiszar A, Sarbandi J, Verl A. Reinforcement Learning of a Robot Cell Control Logic using a Software-in-the-Loop Simulation as Environment. In: *2019 Second International Conference on Artificial Intelligence for Industries (AI4I)*. 2019; pp. 79–84.
- [3] Syed Ahamed MF, Tewolde G, Kwon J. Software-in-the-Loop Modeling and Simulation Framework for Autonomous Vehicles. In: *2018 IEEE International Conference on Electro/Information Technology (EIT)*. 2018; pp. 0305–0310.
- [4] Koenig N, Howard A. Design and use paradigms for Gazebo, an open-source multi-robot simulator. In: *2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (IEEE Cat. No.04CH37566)*, vol. 3. 2004; pp. 2149–2154.
- [5] Coppel Robotics. CoppelSim. o. J.
URL www.coppeliarobotics.com
- [6] Rohmer E, Singh SPN, Freese M. V-REP: A versatile and scalable robot simulation framework. In: *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*. 2013; pp. 1321–1326.
- [7] Mishra R, Javed A. ROS based service robot platform. In: *2018 4th International Conference on Control, Automation and Robotics (ICCAR)*. IEEE. 2018; pp. 55–59.
- [8] Renawi A, Jaradat MA, Abdel-Hafez M. ROS validation for non-holonomic differential robot modeling and control: Case study: Kobuki robot trajectory tracking controller. In: *2017 7th International Conference on Modeling, Simulation, and Applied Optimization (ICMSAO)*. IEEE. 2017; pp. 1–5.
- [9] Takaya K, Asai T, Kroumov V, Smarandache F. Simulation environment for mobile robots testing using ROS and Gazebo. In: *2016 20th International Conference on System Theory, Control and Computing (ICSTCC)*. IEEE. 2016; pp. 96–101.
- [10] Cruz AB, Sousa A, Reis LP. Controller for Real and Simulated Wheelchair With a Multimodal Interface Using Gazebo and ROS. In: *2020 IEEE International Conference on Autonomous Robot Systems and Competitions (ICARSC)*. IEEE. 2020; pp. 164–169.
- [11] Huang Z, Li F, Xu L. Modeling and Simulation of 6 DOF Robotic Arm Based on Gazebo. In: *2020 6th International Conference on Control, Automation and Robotics (ICCAR)*. IEEE. 2020; pp. 319–323.
- [12] Chen J, Deng H, Chai W, Xiong J, Xia Z. Manipulation Task Simulation of a Soft Pneumatic Gripper Using ROS and Gazebo. In: *2018 IEEE International Conference on Real-time Computing and Robotics (RCAR)*. IEEE. 2018; pp. 378–383.
- [13] Weisshardt F, Reiser U, Parlitz C, Verl A. Making High-Tech Service Robot Platforms Available. In: *ISR 2010 (41st International Symposium on Robotics) and ROBOTIK 2010 (6th German Conference on Robotics)*. 2010; pp. 1115–1120.
- [14] Michieletto S, Zanin D, Menegatti E. NAO Robot Simulation for Service Robotics Purposes. In: *2013 European Modelling Symposium*. IEEE. 2013; pp. 477–482.
- [15] Nogueira L. Comparative analysis between gazebo and v-rep robotic simulators. *Seminario Interno de Cognicao Artificial-SICA*. 2014;2014(5).
- [16] Pitonakova L, Giuliani M, Pipe A, Winfield A. Feature and Performance Comparison of the V-REP, Gazebo and ARGoS Robot Simulators. In: *Towards Autonomous Robotic Systems*. 2018; pp. 357–368.
- [17] Santos Pessoa de Melo M, Gomes da Silva Neto J, Jorge Lima da Silva P, Natario Teixeira JMX, Teichrieb V. Analysis and Comparison of Robotics 3D Simulators. In: *2019 21st Symposium on Virtual and Augmented Reality (SVR)*. 2019; pp. 242–251.
- [18] Noori FM, Portugal D, Rocha RP, Couceiro MS. On 3D simulators for multi-robot systems in ROS: MORSE or Gazebo? In: *2017 IEEE International Symposium on Safety, Security and Rescue Robotics (SSRR)*. 2017; pp. 19–24.
- [19] Quigley M, Gerkey B, Conley K, Faust J, Foote T, Leibs J, Berger E, Wheeler R, Ng A. ROS: An open-source Robot Operating System. *ICRA Workshop on Open Source Software*. 2009;3.

- [20] Bihlmaier A, Wörn H. *Hands-on Learning of ROS using Common Hardware*, vol. 625 of *Studies in Computational Intelligence*, pp. 29–50. Springer. 2016;.
- [21] Joseph L, Cacace J. *Mastering ROS for Robotics Programming - Second Edition* -. Birmingham: Packt Publishing, 2nd ed. 2018.
- [22] ros simulation. Simulation Tools In ROS. 2021. GitHub repository.
URL https://github.com/ros-simulation/gazebo_ros_pkgs
- [23] OSFR. Gazebo - A dynamic multi-robot simulator. 2021. GitHub repository.
URL <https://github.com/osrf/gazebo>
- [24] neobotix gmbh. ROS simulation for Neobotix mobile robots. 2020. GitHub repository.
URL https://github.com/neobotix/neo_simulation
- [25] frankaemika. ROS integration for Franka Emika research robots. 2020. GitHub repository.
URL https://github.com/frankaemika/franka_ros
- [26] Siciliano B, Sciavicco L, Villani L, Oriolo G. *Robotics: modelling, planning and control*. Springer Science & Business Media. 2010.
- [27] Ljung L. System identification. *Wiley encyclopedia of electrical and electronics engineering*. 1999;pp. 1–19.
- [28] Gautier M. Numerical calculation of the base inertial parameters of robots. *Journal of robotic systems*. 1991; 8(4):485–506.
- [29] Khalil W, Bennis F. Symbolic calculation of the base inertial parameters of closed-loop robots. *The International journal of robotics research*. 1995; 14(2):112–128.
- [30] Armstrong B. On finding exciting trajectories for identification experiments involving systems with nonlinear dynamics. *The International journal of robotics research*. 1989;8(6):28–48.
- [31] Gautier M, Khalil W. Exciting trajectories for the identification of base inertial parameters of robots. *The International journal of robotics research*. 1992; 11(4):362–375.
- [32] Swevers J, Ganseman C, Tukul DB, De Schutter J, Van Brussel H. Optimal robot excitation and identification. *IEEE transactions on robotics and automation*. 1997;13(5):730–740.
- [33] Khalil W, Dombre E. *Modeling, Identification and Control of Robots*. Elsevier. 2002.
- [34] Vandenberghe L, Boyd S. Semidefinite programming. *SIAM review*. 1996;38(1):49–95.
- [35] Yoshida K, Khalil W. Verification of the Positive Definiteness of the Inertial Matrix of Manipulators Using Base Inertial Parameters. *The International Journal of Robotics Research*. 2000;19(5):498–510.
- [36] Sousa CD, Cortesão R. Physical feasibility of robot base inertial parameter identification: A linear matrix inequality approach. *The International Journal of Robotics Research*. 2014;33(6):931–944.
- [37] Sousa CD, Cortesao R. Inertia Tensor Properties in Robot Dynamics Identification: A Linear Matrix Inequality Approach. *IEEE/ASME Transactions on Mechatronics*. 2019;24(1):406–411.
- [38] Wensing PM, Kim S, Slotine JJE. Linear Matrix Inequalities for Physically Consistent Inertial Parameter Identification: A Statistical Perspective on the Mass Distribution. *IEEE Robotics and Automation Letters*. 2018;3(1):60–67.
- [39] ros industrial. Robotiq. 2019. GitHub repository.
URL <https://github.com/ros-industrial/robotiq>
- [40] IntelRealSense. ROS Wrapper for Intel® RealSense™ Devices. 2021. GitHub repository.
URL <https://github.com/IntelRealSense/realsense-ros>
- [41] SyrianSpock. Intel RealSense Gazebo ROS plugin and model. 2021. GitHub repository, 2020-03-09.
URL <https://github.com/SyrianSpock>
- [42] Rusu RB, Cousins S. 3D is here: Point Cloud Library (PCL). In: *2011 IEEE International Conference on Robotics and Automation*. 2011; pp. 1–4.
- [43] Open Source Robotics Foundation. Gazebo Service Documentation. 2017. ROS documentation, Zugriff: 2021-02-15.
URL https://docs.ros.org/en/kinetic/api/gazebo_msgs/html/index-msg.html
- [44] Moorfeld R. Berechnung und Simulation der Bitfehlerwahrscheinlichkeit von Energiedetektoren bei der Datenübertragung in ultra-breitbandigen (UWB)-Kanälen. Ph.D. thesis, Dresden University of Technology. 2012.
- [45] Zhang H, Udagawa T, Arita T, Nakagawa M. A statistical model for the small-scale multipath fading characteristics of ultra wideband indoor channel. In: *2002 IEEE Conference on Ultra Wideband Systems and Technologies (IEEE Cat. No.02EX580)*. 2002; pp. 81–85.
- [46] Saglam A, Papelis Y. Scalability of Sensor Simulation in ROS-Gazebo Platform with and without Using GPU. In: *2020 Spring Simulation Conference (SpringSim)*. 2020; pp. 1–11.

Virtueller Prüfstand zur simulationsgestützten Bewertung von cyber-physischen Industrie-4.0-Lösungen

Jie Zhang*, Xiaobo Liu-Henke

Ostfalia Hochschule für angewandte Wissenschaften, Fakultät Maschinenbau,
Institut für Mechatronik, Salzdahlumer Str. 46/48, 38302 Wolfenbüttel

*jie.zhang@ostfalia.de

Abstract. Zur Unterstützung der KUM bei Einführung von Industrie-4.0-Lösungen sollen diese mittels fundierter Wissensbasis wissenschaftlich bewertet werden. Deswegen wird eine Simulationsplattform zur Abbildung und Simulation der Produktionssysteme mit verschiedenen Automatisierungsgraden vorgesehen. In diesem Beitrag wird ein virtueller Prüfstand zur Untersuchung der cyber-physischen Produktentstehungssysteme konzipiert und entwickelt. Der virtuelle Prüfstand dient als Simulationsplattform und zugreift auf eine Modellbibliothek für mechatronische Komponenten eines cyber-physischen Produktentstehungssystems zu. Die Simulationsergebnisse des virtuellen Prüfstandes dienen als zuverlässige Wissensbasis für die Bewertung der Einführung von I4.0-Lösungen. Im weiteren Verlauf wird die Anwendung des virtuellen Prüfstandes anhand des Beispiels autonomer Gütertransport in der Intralogistik mittels dezentralen Auftragsmanagements aufgezeigt und validiert.

Einleitung

Im vom Europäischen Fonds für regionale Entwicklung (EFRE) geförderten Verbundprojekt *Methoden und Werkzeuge für die synergetische Konzipierung und Bewertung von Industrie 4.0-Lösungen (Synus)*, an dem insgesamt fünf Professoren der Technischen Universitäten Braunschweig und Clausthal sowie der Ostfalia Hochschule für angewandte Wissenschaften beteiligt sind, wird ein modellgestütztes Werkzeug entwickelt, mit welchem finanzieller Aufwand und Nutzen von Industrie 4.0- (I4.0-)Lösungen in kleinen und mittleren Unternehmen (KMU) bewertet werden sollen, um KMU bei deren Einführung beratend zu unterstützen. Schwerpunkt im Teilprojekt *Modellbasierte Konzeption und Bewertung von Industrie 4.0-Lösungen zur Vernetzung mechatronischer Komponenten in Produktionsanlagen durch Digitalisierung (MiMec)* der Ostfalia Hochschule ist die Mo-

dellierung und Simulation der aus vernetzten mechatronischen Komponenten bestehenden Industrieanlagen und verfügbaren I4.0-Lösungen sowie die systematische Integration fahrerloser Transportfahrzeuge (FTF) durch vollständig digitale Vernetzung in intelligenten, cyber-physischen Produktentstehungssystem (CPPS).

Im vorliegenden Beitrag wird ein virtueller Prüfstand zur simulationsgestützten Bewertung von cyber-physischen I4.0-Lösungen vorgestellt. Dieser virtuelle Prüfstand dient als Simulationsplattform zur Modellierung und Simulation von CPPS mit integrierten I4.0-Lösungen und bietet mit den erzielten Simulationsergebnissen eine wissenschaftliche Grundlage zur qualitativen und quantitativen Bewertung von I4.0-Lösungen für KUM bei der Einführung von solchen Lösungen.

1 Motivation und Zielsetzung

Um die Flexibilität und den Automatisierungsgrad der Herstellungsindustrie weiter zu erhöhen und damit deutsche Unternehmen im globalen Wettbewerb weiterhin einen Vorsprung aufrecht zu erhalten, wurden schon einige intelligente Lösungsansätze entwickelt und auf dem Markt zur Verfügung gestellt [1]. Abbildung 1 stellt einen Überblick über ein CPPS unter Einsatz von I4.0-Lösungen vor. Die einzelnen Produktionsakteure z.B. Produktionsmaschinen, Lager, FTF usw. in einer Produktionsanlage können durch Integration von intelligenter Hardware z.B. Smarter Sensoren und Aktoren, Kommunikationstechnologie z.B. WLAN und RFID, intelligenten Funktionen mittels KI, usw. ihre Umgebung erkennen, direkt mit anderen Produktionsakteuren kommunizieren sowie autonom reagieren. Damit erfüllt die Produktionsanlage durch schnelle Kommunikation, Berechnung und Regelung die Anforderungen von Industrie-

4.0, Digitalisierung, Internet of Things (IoT) und vielen weiteren innovativen Technologien in diesem Zusammenhang.

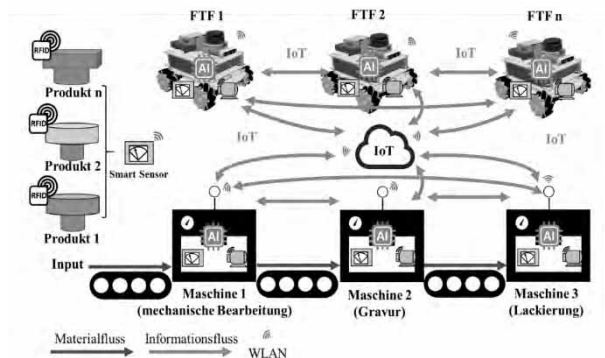


Abbildung 1: Cyber-physisches Produktentstehungssystem durch Integration von I4.0-Lösungen

Aber Unternehmen, insbesondere KMU, sind bisher zurückhaltend bei der Einführung und Nutzung solchen digitalen Technologien [2]. Einerseits ist diese Unsicherheit auf mangelnde Kenntnisse der verfügbaren Technologien zurückzuführen, andererseits ist es bisher für Unternehmen kaum möglich, die Potenziale und Konsequenzen der Einführung von I4.0-Lösungen abzuschätzen.

Um eine fundierte Wissensbasis für KMU zu etablieren und damit für qualitative sowie quantitative Bewertungen von interdisziplinären I4.0-Lösungen zu verfügen, wird im Rahmen des Forschungsprojektes Synus ein virtueller Prüfstand vorgesehen. Mit dessen Hilfe können Produktionsanlagen mit verschiedenen Automatisierungsgraden nach Kundenwunsch individuell abgebildet und simuliert werden. Das heißt, mittels virtuellen Prüfstands können die konventionellen bzw. automatisierten und auch autonomen Produktionsanlagen unter Einsatz von I4.0-Lösungen mit gegebenen Randbedingungen modelliert und simuliert werden. Die Ergebnisse lassen sich miteinander vergleichen und nach wissenschaftlicher Analyse als zuverlässige Grundlage für Bewertungen der eingeführten I4.0-Lösungen dienen. Außerdem liegt ein weiterer Vorteil des virtuellen Prüfstandes für die Entwickler der I4.0-Lösungen insbesondere darin, dass nicht nur die modellbasiert entwickelten intelligenten Funktionen der Produktionsakteure, sondern auch die gesamten Produktionsanlagen im Entwicklungsprozess mittels Model-in-the-Loop- (MiL-) und Software-in-the-Loop- (SiL-)Simulation ohne reale Forschungsträger oder Prototypen in einer realitätsnahen virtuellen Simulationsumgebung abgesichert und optimiert werden können.

2 Stand des Wissens

Der Begriff Industrie-4.0 wurde erstmal 2011 auf der Hannover Messe genannt und bezeichnet den Wandel der Industrie zur digitalisierten, vernetzten Produktion [1]. Dieser Wandel war oftmals so ausschlaggebend, dass hierbei von einer Revolution die Rede ist. Dabei wird die Produktion als komplexes informationsverarbeitendes System verstanden, wobei die Verwendung von Informations- und Kommunikationstechniken elementarer Bestandteil ist. Maschinen, die von kleinen, integrierten Systemen – sog. Embedded Systems – gesteuert werden, werden umfassend nicht nur mit dem gesamten Unternehmensnetz verbunden, sondern in ein weltumspannendes Kommunikationsnetz, dem Internet der Dinge (IoT), eingebunden. Hierdurch entstehenden sog. cyber-physische Systems (CPS), die die Verschmelzung der realen mit der virtuellen Welt beschreiben [3].

In den letzten Jahren wurden in zunehmendem Maße intelligente Technologien im Kontext von I4.0 entwickelt und veröffentlicht. Mittels eines Klassifikationsmodells werden die I4.0-Lösungen nach den Wirkungsfeldern identifiziert und charakterisiert [4]. Nach wissenschaftlicher Analyse lässt es sich festlegen, dass die auf dem Markt verfügbaren I4.0-Lösungen fallspezifisch entwickelt und deren Wirksamkeit auch durch Technologiereifegrad, organisatorische Randbedingungen und andere Faktoren begrenzt wurden. Z.B. sind Werkzeuge und Algorithmen zur Erfassung und Analyse der Betriebsdaten und intelligente Funktionen mittels KI oder Machine Learning zur Erkennung der Schadenfälle die grundlegenden Voraussetzungen von Predictive Maintenance, welches das Vorhersagen des Wartungsbedarfs durch frühzeitige Erkennung von Anomalien forciert [5]. Es sei auch auffällig, solche I4.0-Lösungen sind bereichsübergreifend und können in unterschiedlichen Unternehmensbereichen und Abschnitten des Produktentstehungssystems eingesetzt werden.

Um die Auswirkungen der eingesetzten I4.0-Lösungen auf das Produktionssystem zu bewerten, ist es notwendig, das Produktionssystem mit und ohne I4.0-Lösungen zu simulieren und die Systemperformance durch eine Reihe von Prozesskennwerten zu quantifizieren. Durch den Vergleich der Prozesskennwerte lässt sich die Aussage der Einführung von I4.0-Lösungen für KUM ziehen. Aber bislang gibt es für CPPS mit I4.0-Lösungen jedoch keine Möglichkeit, komplexe Testszenarien mit mehreren Teilnehmern realitätsnah in einer virtuellen Plattform abzubilden und zu

simulieren. Dies führt zu einer Lücke in der systematischen Bewertung der I4.0-Lösungen.

3 Methodik

Wie bereits erwähnt, werden I4.0-Lösungen fallspezifisch entwickelt. Durch vielfältige Faktoren, insbesondere zur Integration der Lösungen in bestehende Produktionsanlagen, ist daher deren Wirksamkeit begrenzt. Zur Integration ist eine ganzheitliche Betrachtung des Gesamtsystems mit einer strukturierten Vorgehensweise für den systematischen Entwurf erforderlich. Die I4.0-Lösungen sollen nicht nur auf Aggregationsebene, z.B. Produktionsmaschine und Transportfahrzeug, sondern auch auf Komponentenebene, z.B. Sensor und Aktor, integriert werden. Um die Flexibilität und den Automatisierungsgrad des Produktionssystems umfangreich zu erhöhen, müssen die Schnittstellen zum Informationsaustausch sowie die physikalische Kopplung zwischen Komponenten definiert werden. Außerdem ermöglicht eine klare Strukturierung des Gesamtsystems die Beherrschung der Komplexität bei der Integration vernetzten mechatronischen Systeme mittels I4.0-Lösungen.

Modularisierung und Hierarchisierung stehen im Mittelpunkt zur Strukturierung. Bei der Modularisierung werden in einem Top-Down-Prozess Teilfunktionen mit mechatronischen Komponenten aus dem Gesamtsystem abgeleitet und in Module gekapselt. Bei der Hierarchisierung erfolgt die Anordnung der Teilfunktionen in vier Ebenen mit definierten Schnittstellen, wie in der folgenden Abbildung 2 dargestellt [6]:

- **Mechatronisches Funktionsmodul (MFM):** Die unterste Hierarchieebene bilden die MFM, bestehend aus nicht weiter teilbaren mechatronischen Systemen inklusive Grundaufbau, Sensorik, Aktorik und Informationsverarbeitung. Jedes gekapselte MFM besitzt eine definierte Funktionalität und beschreibt die Dynamik des Systems.
- **Mechatronische Funktionsgruppe (MFG):** Durch Kopplung mehrerer MFM und Hinzufügen einer Informationsverarbeitung entstehen MFG. MFG ermöglichen durch Nutzung der unterlagerten MFM mit deren Aktorik die Realisierung höherwertiger Funktionen wie die integrierte Dynamikregelung, die Trajektorienplanung oder das Energiemanagement.
- **Autonomes Mechatronisches System (AMS):** Das mechatronische Gesamtsystem bildet die nächste Hierarchieebene der AMS durch Kombination von mehreren MFG, welche in diesem Beispiel durch die autonome Produktionsmaschine selbst oder autonome Transportfahrzeuge repräsentiert werden. Nach Verarbeitung der zur Verfügung stehenden Informationen von mechanischer Tragstruktur und Sensorik erteilt entsprechende Befehle an unterlagerte MFG und MFM. Das AMS verfügt auch über informationstechnische Schnittstellen zu weiteren Systemen.
- **Vernetztes Mechatronisches System (VMS):** Werden mehrere AMS nebeneinander betrieben, z.B. um einen Kundenauftrag zu verarbeiten, so bedarf es einer höheren Instanz zur Koordination. Diese rein informationstechnische Kopplung auf oberster Ebene ist ein VMS bzw. in diesem Beispiel ein CPPS, welche einer I4.0-Produktionsstraße oder einer I4.0-Fabrik entspricht. Durch das übergeordnete CPPS

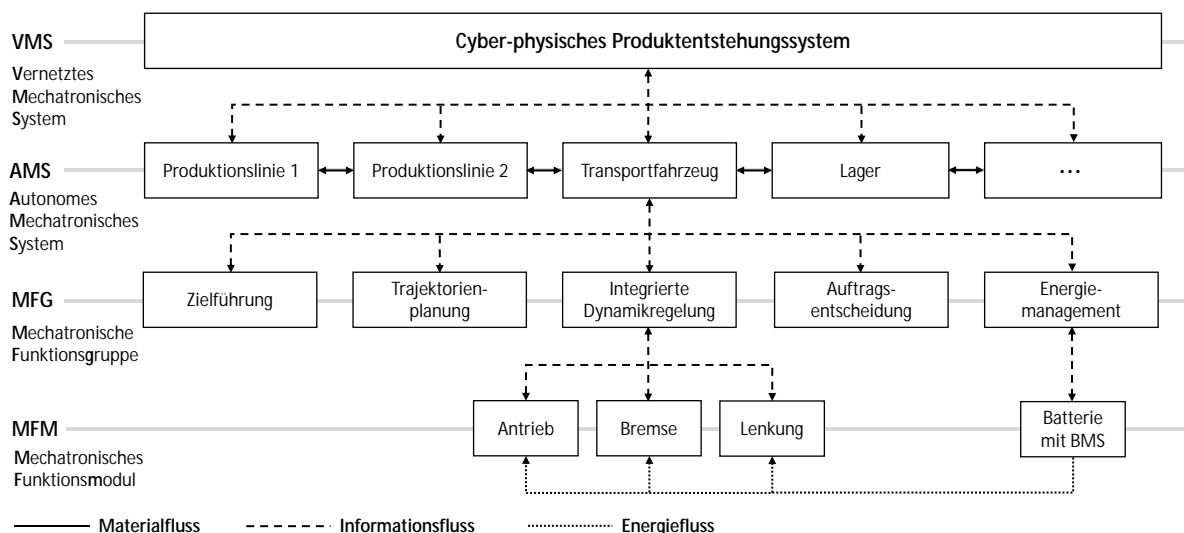


Abbildung 2: Hierarchische Strukturierung eines CPPS

werden die Produktinformationen verwaltet und z. B. per WLAN an alle relevanten Komponenten der Produktionsstraße weitergegeben.

4 Konzeption des virtuellen Prüfstandes

In diesem Abschnitt werden die Anforderungen an den virtuellen Prüfstand aus dem Ziel des Forschungsprojektes Synus ermittelt und beschrieben. Aufbauend auf den Anforderungen wird der virtuelle Prüfstand konzipiert und in folgenden vorgestellt.

4.1 Anforderungen an den virtuellen Prüfstand

Die Anforderungen an den virtuellen Prüfstand werden aus dem Ziel des Forschungsprojektes abgeleitet (vgl. Kapitel 1). Das CPPS mit verschiedenen Automatisierungsgraden soll mittels virtuellen Prüfstandes abgebildet und simuliert werden. Die Simulationsergebnisse sollen für die quantitative sowie qualitative Bewertung von I4.0-Lösungen verfügbar sein. Hieraus ergeben sich im Detail folgende Anforderungen:

- Die in eine Produktionsanlage integrierten I4.0-Lösungen können auf verschiedenen Ebenen bzw. auf Systemebene, welche verschiedene Produktionsakteure beinhaltet z.B. Produktionsmaschinen zur Materialbearbeitung, FTF zum Gütertransport usw., und

auf Komponentenebene z.B. Smart Aktor und Sensor abgebildet und simuliert werden.

- Zur realitätsnahen Simulation lassen sich vielfältige Szenarien realitätsnah hoch automatisiert generieren.
- Der virtuelle Prüfstand soll vielfältige Modelle zur Abbildung der mechatronischen Komponenten in einer Produktionsanlage nach Kundenwunsch anbieten. Die Modelle sollen wiederverwendbar, parametrierbar und rekonfigurierbar sein.
- Die Simulationsergebnisse sollen übersichtlich und verständlich aufbereitet werden. D.h. die Simulationsergebnisse sollen in 2D-Form z.B. Tabelle und Diagramm und auch in 3D-Form z.B. Animation visualisiert werden.

4.2 Architektur des virtuellen Prüfstandes

Anhand der bereits erwähnten Anforderungen wird der virtuelle Prüfstand konzipiert. Abbildung 3 gibt einen Überblick über dessen Konzept. Im Detail werden die folgenden Teilmodule des virtuellen Prüfstandes vorgestellt:

- Anforderungsmanagement:** Durch das Anforderungsmanagement können die konkreten Anforderungen bzw. die Lastenheft und Spezifikationen mithilfe der hierarchischen Strukturierung auf Systemebene und dann auch auf der Komponentenebene konkret definiert [7]. Anhand solcher Anforderungen können entsprechende Testszenarien konstruiert

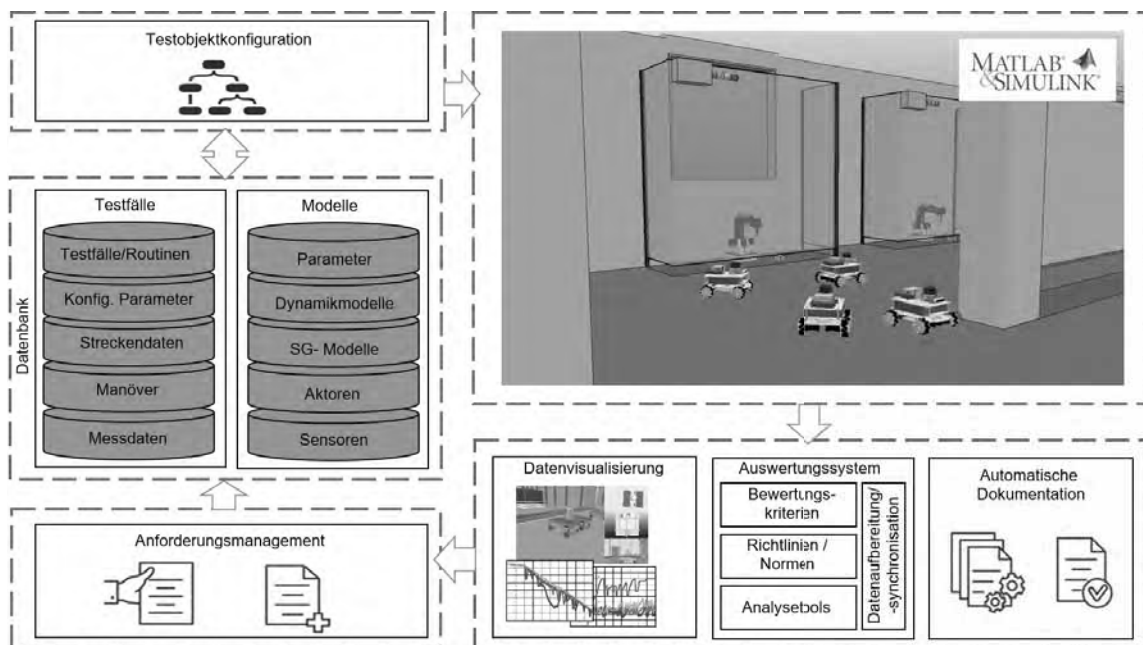


Abbildung 3: Konzept vom virtuellen Prüfstand zur Untersuchung des CPPS

und auch die Modelle aus der Datenbank ausgewählt werden.

- **Datenbank:** Die Datenbank ist ein Kernelement des virtuellen Prüfstandes und bestellt aus den beiden Teilen Szenarien- und Modellbibliothek. Durch den Szenariogenerator lassen sich die Testszenarien konstruieren, bei dem die Produktionsanlage bzw. Anzahl, Art und auch Funktionalität der jeweiligen Produktionsakteure ausführlich beschrieben werden. Die Szenariobeschreibung können in verschiedenen Datenformaten z.B. textuell und durch Excel-Datei formuliert werden. Um die Wiederwendbarkeit, Parametrierbarkeit und Rekonfigurierbarkeit der Modelle zu gewährleisten, werden die mechatronischen Modelle einer Produktionsanlage auf verschiedenen Ebenen z.B. die Funktionsmodule von autonomen FTF auf Systemebene und Sensoren und Aktoren von autonomen FTF auf Komponentenebene, nach der hierarchischen Strukturierung mit definierten Schnittstellen moduliert und in einer Modellbibliothek angeordnet, die für die Abbildung einer Produktionsanlage verfügen.
- **Testobjektconfiguration:** Bei Testobjektconfiguration werden die Modelle aus Modellbibliothek über Schnittstelle miteinander integriert und anschließend nach der Szenariobeschreibung mit den Testszenarien integriert, um den Informations-, Material- und Energiefluss einer Produktionsanlage abzubilden. Die Simulation des konfigurierten Modells erfolgt mittels CAE-Werkzeugs Matlab/Simulink.
- **Datenvisualisierung und Dokumentation:** Obwohl mit den eingebauten Funktionen von Matlab/Simulink die Simulationsergebnisse in 2D-Form von Diagrammen und Tabellen formuliert werden können, ist es nicht intuitiv. Z.B. kann die Interaktion zwischen FTF und Produktionsmaschinen nicht direkt über solche Diagramme und Tabellen nachvollzogen werden. Deswegen wird im virtuellen Prüfstand eine Software zur 3D-Visualisierung der Produktionsanlage integriert. Die Simulationsergebnisse werden auch als Dokumente archiviert, um sie wieder bei Bedarf abzurufen.
- **Auswertungssystem:** Im virtuellen Prüfstand wird die im Rahmen von Synus-Projekt entwickelten Bewertungsmethodik verwendet, um die I4.0-Lösungen qualitativ und quantitativ zu bewerten. Nach der Zielsetzung der eingesetzten I4.0-Lösungen werden

die relevanten Zielkriterien aus verschiedenen Bewertungsperspektiven ermittelt und die Prozesskennzahl zur Quantifizierung der Zielerreichung aus den Simulationsergebnissen erfasst [8]. Solche Prozesskennzahl wird nach wissenschaftlicher Analyse verarbeitet und ist zur Bewertung verfügbar.

5 Simulation des autonomen Gütertransportes mittels virtuellen Prüfstandes

In diesem Abschnitt wird die Anwendung des virtuellen Prüfstandes anhand des Beispiels von autonomem Gütertransport in einem Labortestfeld mittels dezentralen Auftragsmanagements aufgezeigt und validiert.

Für die realitätsnahe Entwicklung und Erprobung der Funktionalitäten eines einzelnen FTF und der IoT-basierten Kommunikation in Produktionsanlagen unter Echtzeitbedingungen wird ein cyber-physisches Labortestfeld konzipiert und entwickelt [9]. Abbildung 4 zeigt einen Überblick über das cyber-physische Labortestfeld, welches aus autonomen FTF zum Gütertransport, Produktionsmaschinen für Transformation, Lager für Materiallagerung und -bereitstellung und Warenausgang als Senke für Güter besteht. Um die modellbasiert entwickelten Funktionen zur Realisierung des autonomen Gütertransportes frühzeitig abzusichern, wird das Labortestfeld im virtuellen Prüfstand abgebildet und simuliert.

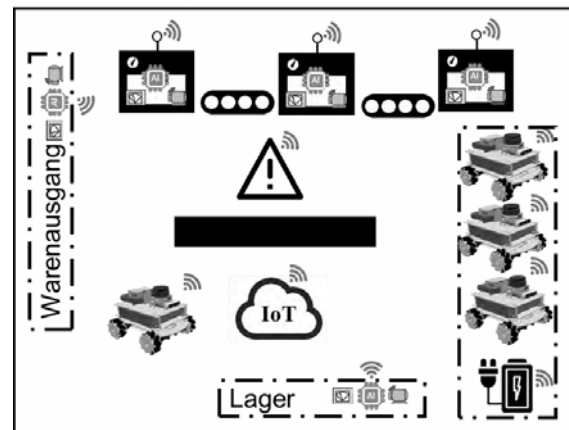


Abbildung 4: Cyber-physisches Labortestfeld [9]

5.1 Szenariobeschreibung

Bei der Simulation wird zuerst nach den Anforderungen an autonomen Gütertransport das Simulationsszenario

beschreiben. Bei der Beschreibung des Simulationsszenarios sollten die Art der Produktionsakteure z.B. FTF, Produktionsmaschinen, Lager usw., deren Anzahl und Funktionalität detailliert angegeben werden.

Wie in der Abbildung 4 veranschaulicht, wird das Labortestfeld durch die folgenden Produktionsakteure im Detail realisiert [9]:

- 3 Produktionsmaschinen für Transformation
- 4 autonome FTF zum Gütertransport
- 2 Lager für Materiallagerung und -bereitstellung
- 1 Warenausgang als Senke für Güter

Alle Produktionsakteure können mittels IoT-basierter Kommunikation direkt Informationen austauschen und hierdurch mithilfe der eigenen Intelligenz selbst Entscheidungen treffen [6, 10]. D.h. gegenüber der konventionellen Intralogistik wird die sogenannte zentrale Leitsteuerung dezentralisiert. Der grundlegende Unterschied zwischen zentraler Leitsteuerung und dezentralem Auftragsmanagement wird durch die Abbildung 5 klar veranschaulicht. Die obere Abbildung zeigt die Funktionsweise der zentralen Leitsteuerung als ein Kernelement zur Kopplung des Produktionssystems und automatisierten FTF, während die untere Abbildung stellt dar, dass beim dezentralen Auftragsmanagement alle Produktionsakteure durch IoT-basierte Kommunikation zur Ausführung der Transportaufträge miteinander vernetzen. Das

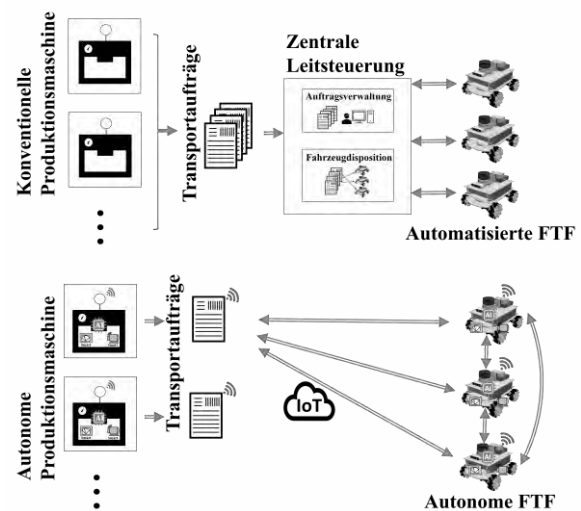


Abbildung 5: Zentrale Leitsteuerung der konventionellen Intralogistik (oben) und dezentrales Auftragsmanagement für autonome FTF (unter)

dezentrale Auftragsmanagement wird zur optimalen Auftragsverteilung hinsichtlich Energieverbrauch und Transportzeit für autonomen FTF konzipiert und entwickelt.

Anhand der Transportaufträge, die von den Produktionsmaschinen erstellt und anschließend über die IoT-basierte Kommunikation z.B. WLAN den autonomen FTF mitgeteilt werden, können die autonomen FTF eigenständig eine Route zwischen der aktuellen Position und Zielposition planen und für diese Route die Transportzeit und

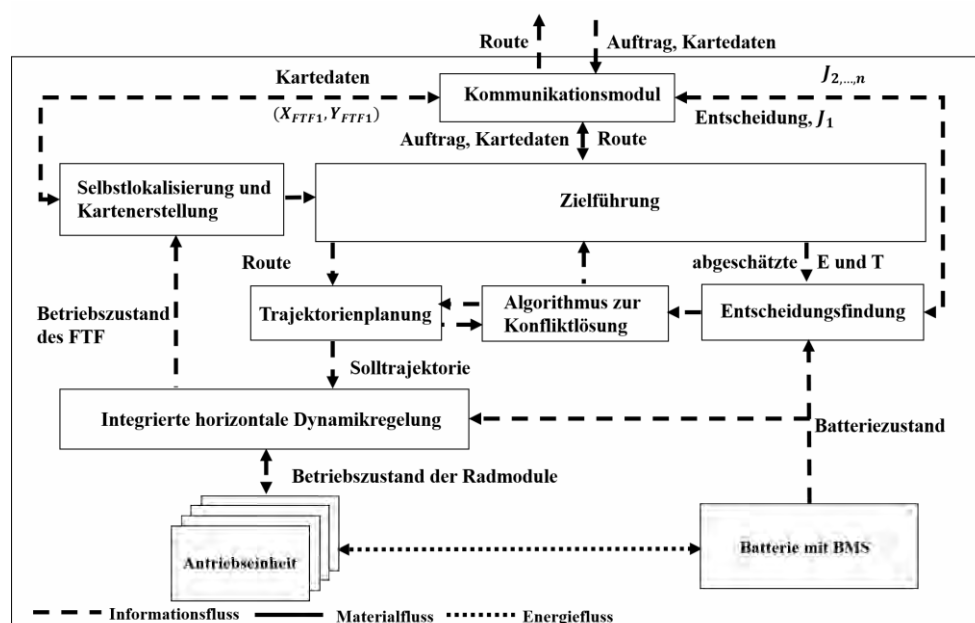


Abbildung 6: Funktionsstruktur eines autonomen FTF zur Realisierung des autonomen Gütertransportes

der Energieverbrauch abschätzen. Damit werden die Fahrtkosten für den Transportauftrag (TA) mithilfe der künstlichen Intelligenz bestimmt und durch das Kommunikationsmodul direkt miteinander ausgetauscht, sodass eine gemeinsame Entscheidungsfindung, welches FTF diesen TA übernehmen und durchführen soll, erfolgt.

Nach der Szenariobeschreibung werden die notwendigen Funktionsmodule der autonomen FTF zur Realisierung des autonomen Gütertransportes aus der Modellbibliothek ausgewählt. In Abbildung 6 wird die Funktionsstruktur der Informationsverarbeitung eines autonomen FTF nach der hierarchischen Strukturierung dargestellt, welche die Schnittstellen zwischen einzelnen Funktionsmodulen definiert und selbstständige Entscheidungsfindung zur Übernahme der TA und damit Ausführung der TA. Die einzelnen Funktionen werden in [6] und [11] ausführlich beschrieben.

5.2 Simulationsdurchführung und -ergebnisse

Das Labortestfeld wird nach dem Konzept im virtuellen Prüfstand abgebildet, wie in der Abbildung 7 veranschaulicht. Die benötigten Funktionsmodule werden nach der Szenariobeschreibung aus der Modellbibliothek ausgewählt und parametrisiert. Hinsichtlich der definierten Schnittstellen integrieren sich die einzelnen Funktionsmodule miteinander und in CAE-Werkzeug Matlab/Simulink das Simulationsmodell gebildet.

Die Simulation des dezentralen Auftragsmanagements wird nach der Szenariobeschreibung durchgeführt. Die Simulationsergebnisse werden in der folgenden Abbildung 7 und Tabelle 1 zusammengefasst. TA werden von der Produktionsmaschine zu verschiedenen Zeiten während des Produktionsprozesses generiert. Abbildung

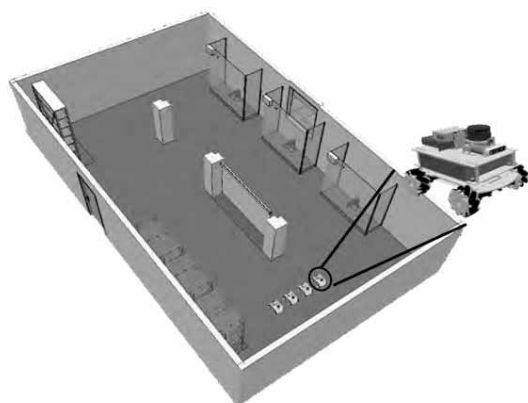


Abbildung 7: Labortestfeld im virtuellen Prüfstand

8 zeigt die individuelle Auftragsverteilung an jeweilige FTF durch das dezentrale Auftragsmanagement. Es braucht insgesamt 34,591s die 4 TA von Produktionsmaschinen auszuführen. In der Tabelle 1 werden die Transportzeit und die Länge des Fahrweges als Maßstäbe von Energieverbrauch des jeweiligen TA zusammengefasst.

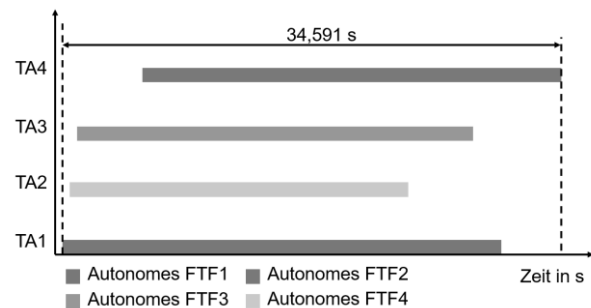


Abbildung 8: Auftragsdisposition durch dezentrales Auftragsmanagement

	Transportzeit [S]	Fahrweg [m]
TA1	29,981	21,2473
TA2	23,501	14,2763
TA3	27,511	18,2809
TA4	29,091	19,8661

Tabelle 1: Transportzeit und Länge des jeweiligen Fahrweges von autonomen FTF

6 Zusammenfassung und Ausblick

Im vorliegenden Beitrag wird das Konzept eines virtuellen Prüfstands zur simulationsgestützten Bewertung der cyber-physischen I4.0-Lösungen vorgestellt. Ausgehend von dem Ziel des Synus-Projektes zur Bewertung der I4.0-Lösungen ist ein virtueller Prüfstand als eine Simulationsplattform erforderlich, der für Abbildung und Simulation der CPPS mit und ohne I4.0-Lösungen zur Verfügung steht. Aufbauend auf den Anforderungen an den virtuellen Prüfstand werden dessen Struktur und Module abgeleitet und konzipiert. Die Anwendung des virtuellen Prüfstandes wird durch die Simulation des dezentralen Auftragsmanagements von autonomen FTF in Labortestfeld aufgezeigt und validiert.

In den nächsten Arbeitsschritten sollen die Modellbibliothek um weitere mechatronischen Komponenten und Funktionsmodule erweitert werden, um über umfangreiche Möglichkeiten für Modellierung und Simulation der I4.0-Lösungen auf dem Markt zu verfügen.

Danksagung

Dieser Beitrag wurde im Rahmen des Teilprojekts *Modellbasierte Konzeption und Bewertung von Industrie 4.0-Lösungen zur Vernetzung mechatronischer Komponenten in Produktionsanlagen durch Digitalisierung (MiMec)* des Verbundprojekts *Methoden und Werkzeuge für die synergetische Konzipierung und Bewertung von Industrie 4.0-Lösungen (Synus)* durch den Europäischen Fonds für regionale Entwicklung (EFRE) unter dem Förderkennzeichen ZW 6 85012454 gefördert. Die Verantwortung für den Inhalt liegt bei den Autoren.



EUROPÄISCHE UNION
Europäischer Fonds für
regionale Entwicklung



Literatur

- [1] Bundesministerium für Wirtschaft und Energie. *Impulse für die Digitalisierung der deutschen Wirtschaft - Digitale Agenda des BMWi*. 2015. [online] Available: <https://www.bmwi.de/Redaktion/DE/Publikationen/Digitale-Welt/impulse-digitalisierung-deutsche-wirtschaft.pdf>
- [2] McKinsey&Company. *Industry 4.0. How to navigate digitization of the manufacturing sector*. 2015.
- [3] Das Technologie-Netzwerk, Ed. *Auf dem Weg zu Industrie 4.0: Lösungen aus dem Spitzencluster It's Owl*. 2016. [online] Available: https://www.its-owl.de/fileadmin/PDF/Industrie_4.0/Auf_dem_Weg_zu_Industrie_4.0_Loesungen_aus_dem_Spitzencluster_its_OWL_RGB.pdf
- [4] Inkermann D, Lembeck H, Schneider D, Zhang J, Martin N, Thiede S. An integrated framework to classify Industry 4.0 technologies within production and product development. *29th CIRP Design Conference*; 2019 May; Póvoa de Varzim, Portugal.
- [5] Kagermann H, Wahlster W, Helbig J. *Umsetzungsempfehlungen für das Zukunftsprojekt Industrie-4.0*. Abschlussbericht des Arbeitskreises Industrie 4.0, 2014.
- [6] Zhang J, Yarom OA, Liu-Henke X. Decentralized, self-optimized order-acceptance decision of autonomous guided vehicles in an IoT-based production facility. *The 9th International Conference on Mechatronics and Control Engineering*; 2020 July; Rome, Italy.
- [7] Yarom OA, Zhang J, Raulf C, Liu-Henke X, Vietor T. Anforderungsmanagement für die modellbasierte Entwicklung mechatronischer Systeme im digitalisierten und vernetzten Umfeld. *Workshop ASIM*; 2020 Oktober; Sankt Augustin, Germany.
- [8] Lembeck H, Schneider D, Essakly A, Inkermann D, Spengler T, Vietor T. Bewertungsmethodik für die Prozessintegration von Industri 4.0-Lösungen in die Produktentwicklung. *Stuttgarter Symposium für Produktentwicklung SSP 2019*; 2019 Mai; Stuttgart, Deutschland.
- [9] Liu-Henke X, Jacobitz S, Göllner M, Zhang J, Scherler S, Yarom O. Cyber-physical industry 4.0 test field to simulate self-optimizing intralogistics. *19th IEEE-Mechatronika 2020*; 2020 December; Prague, Czech Republic.
- [10] Liu-Henke X, Yarom OA, Göllner M. Concept for the simulation-based description of intelligent mechatronic components and their networking in an I4.0 production line. *ICEIME Conference*; 2019 August; Prague, Czech.
- [11] Zhang J, Liu-Henke X. Konfliktfreie, selbstoptimierte Trajektorienplanung für ein fahrerloses Transportfahrzeug zur Durchführung des autonomen Gütertransportes im Produktionsumfeld. *Workshop ASIM*; 2020 Oktober; Sankt Augustin, Germany. ARGESIM Report 59 (ISBN 978-3-901608-93-3). 259-266. DOI: 10.11128/arep.59.a59037

Simulation Model of a Torque-Limiting Clutch with Adjustable Design Parameters to Investigate the Release Behavior

Michael Steck¹, Daniel Paland¹, Thomas Gwosch¹, Sven Matthiesen^{1*}

¹IPEK – Institut für Produktentwicklung, Karlsruher Institut für Technologie (KIT), Kaiserstr. 10, 76131 Karlsruhe, Deutschland; **sven.matthiesen@kit.edu*

Abstract. For the design of torque-limiting clutches in the early phases of product development, simulation models are necessary which allow an adjustment of the release behavior by defining the design. Known and accessible simulation models do not allow profile variation of torque-limiting clutches.

Based on an analysis of a torque-limiting clutch, quantitative relationships between the design parameters and the release behavior were investigated and implemented in a simulation model. The developed model represents the torque defined by the profile as a function of the angular position and the direction of rotation. The simulation model is parameterized for instance via the geometry of the profile and the spring stiffness. The model takes into account friction behavior across all components.

With this model, the behavior of the torque-limiting clutch can be simulated and an adaptation of the shape of the clutch to its function can be estimated. Verification results show that the model can be used for early design and application in testing.

Introduction

In product development, design parameters are defined in order to realize a desired function [1]. This requires an understanding of the relationship between design parameters and the resulting behavior.

For the derivation of target values of the design parameters (described in the following as design targets) and the testing of development variants during development, both test benches and simulations are used. Increasingly, simulations are used to predict design parameters. Simulations are used to build up an understanding of the system and to define design targets. This is helpful if subsystems are not yet physically available and only simulation models can be used for early validation in the development process. The key is not only to be able to investigate principles, but also to map design details in

the simulation model in order to enable a preliminary assessment of the components and to define design targets. Such simulations exist for the testing of subsystems in the automotive, aerospace and electrical equipment industries, among others. The simulation models of the subsystems provides the basis for investigations on test benches. Particularly for safety-critical components, such as overload or torque-limiting clutches, knowledge of the relationships [2; 3] between design and function is important for preliminary design.

Challenge. In the state of research, simulation models of overload clutches already exist [2–6]. The available models mainly represent clutches with notches. Other shape variants are rarely considered by simulations in commercial programs.

These existing models can only be used to a limited extent for the prediction of profile shape variants of the existing torque-limiting clutch. It is not possible to adapt the available models for use in prediction and bench testing. Known and accessible models do not fully represent the relationship between design parameters and release behavior. The development of a simulation model of the existing torque-limiting clutch with variable design parameters is necessary.

Aim. The aim of this paper is to develop a simulation model for mapping the torque-limiting clutch release behavior. The relationship between design parameters and release behavior is analyzed on the basis of an existing torque-limiting clutch, which is designed as a locking element clutch. The determined relationships are transferred to a simulation model and the functionality is compared with experiments on a test bench. The modeling is exemplified by the torque-limiting clutch of a cordless drill/driver of the type GSR 10 8-2-LI Plus (Robert Bosch GmbH), see Figure 1 and 2.

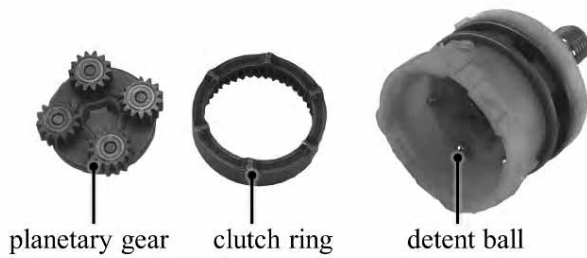


Figure 1: The exemplified system of the mechanical torque-limiting clutch consisting of a clutch ring with detents and the planetary gear. [7]

1 Materials and Methods

For model building, a three-step procedure based on [8–10] with system analysis, implementation and validation is used.

Clutch. The clutch (see Figure 2) limits the torque mechanically. The clutch consists of detents on a clutch ring, which is integrated into the third stage of the planetary gear unit. The detent balls are pressed onto the clutch ring by springs. The detent balls and springs are guided in the clutch housing, where they can support the applied torque. When the torque applied from outside (output shaft) increases, the springs are further compressed, causing the detent balls to move up along the detents. As soon as the detent balls move above the detents, no more torque can be supported. The clutch then releases and limits the maximum transferable torque of the output shaft to a defined value. This torque limit can be adjusted via the preload of the springs. The components of the clutch are shown in Figure 1 and Figure 2.

1.1 System Analysis

The basis for modeling is an analysis of the existing clutch. For this purpose, the clutch is broken down into individual subsystems and the relevant parameters are identified. In addition, the relationships between design parameters and behavior are derived in the form of equations. These relationships in the form of equations are based on a force equilibrium. The relationships and the defined parameters are then used in the model building.

Based on an analysis of the clutch, quantitative relationships are derived.

1.2 Implementation

Based on the analysis, the physical simulation model is

implemented and parameterized as a grey box model.

The model building and simulation program used is Matlab Simulink and Simscape. The determined relationships are implemented in formulas and relationships within a new Simscape component or Simulink block.

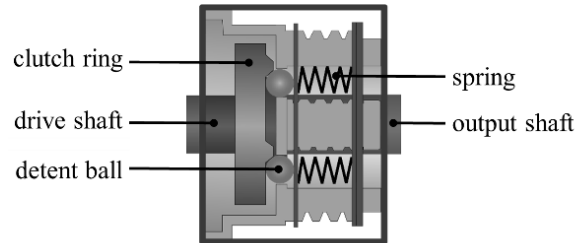


Figure 2: Representation of the components of the planetary gear with clutch [11]. The spring force forces the detent balls against the detent on the clutch ring to prevent them from rotating.

For parameterization, the geometry and the relevant parameters of the clutch are recorded. The spring stiffness parameters is determined using a tensile testing machine. The profile geometry is recorded via a digital microscope with 3D measuring function. The geometry of the torque-limiting clutch is mapped via the profile shape. The occurring friction was summarized in two parameters. The friction was assumed to be sliding friction. The friction parameters are not precisely assigned to the friction contact. For the simulation, it is assumed that the friction conditions do not change significantly at different clutch stages and operating points.

1.3 Test Setup for Validation

The simulation model is validated with measurement data from a test rig, with actuators and measuring shafts (sensor 1 and sensor 2) on the input and output sides (see Figure 2). The validation of the virtual clutch is carried out in comparison with mechanical experiments at different operating points (defined by the preload levels 3, 5, 10, 15) on a test bench.

The verification is performed on the powertrain test bench (see Figure 3). For this purpose, the clutch between two load machines (actuator 1 and actuator 2) is investigated. In order to evaluate the performance of the clutch, the time signals are compared with the simulation results. The test setup is shown in Figure 3.

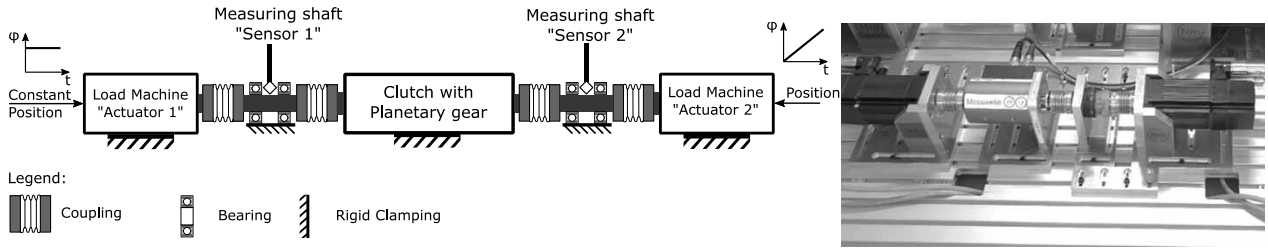


Figure 3: Test bench for comparison with simulation. The test bench contains the clutch between two load machines and additional measuring shafts.

In order to achieve a high degree of comparability, the relevant properties of the test bench are taken into account in the simulation. Thus, the inertias and stiffnesses of the test bench components such as measuring shafts and connecting elements are integrated into the simulation.

For this purpose, the mechanical properties can be implemented in the simulation based on data sheets. The evaluation is based on characteristic points in the time course of torque and speed. The maximum clutch torque and the negative torque are used as evaluation variables. The comparison is made by evaluating the torque and angular position signals at the interfaces of the clutch. In order to ensure simple conversion of the time curves, angle inputs were used in addition to speed inputs. For a speed-independent representation, the results are also plotted against the angular position of the clutch. For this purpose, the angular positions are used to infer the current position of the clutch. The time curve can then be converted into an angular curve. This allows the function of the clutch to be assigned to individual sections of the elevation. The relationship between design and function thus becomes notable.

2 Results

First, the system is presented and the analysis results are shown. Then, the implementation of the relationships of the model is shown and the simulation results are compared to experiments.

2.1 System Analysis

The torque-limiting clutch of the cordless drill is integrated in a planetary gear. The clutch limits the torque mechanically and consists of six balls mounted in the plastic housing, which prevent the clutch from rotating via a profile. The design parameters (geometry, material) of the clutch ring, the springs and the detent balls define

the functional properties of the clutch and thus the behavior with regard to the safety function. It is assumed that the detent balls are always in contact with the clutch ring.

In addition to the profile shape, the properties of the spring and the ball are important for the function. The torque depends on the position of the ball on the detent. The clutch torque was determined based on a balance of forces. The gradient at the contact point is determined by the position of the ball on the profile. The torque for the individual angular positions is determined via a following equation.

$$M_p(\varphi) = r_p F_c \left(\frac{\frac{\Delta z}{r_p \Delta \varphi} + \mu_k \operatorname{sign}(\dot{\varphi})}{1 - \mu_k \operatorname{sign}(\dot{\varphi}) \frac{\Delta z}{r_p \Delta \varphi}} \right) \quad (1)$$

$$M_s(\varphi) = r_p F_c \mu_s \operatorname{sign}(\dot{\varphi}) \quad (2)$$

$$M_c(\varphi) = M_p + M_s \quad (3)$$

$M_c(\varphi)$ describes the torque of the entire clutch, $M_p(\varphi)$ the torque generated by the force redirection and the profile friction, $M_s(\varphi)$ the torque of the friction housing, r_p the radius of the profile track, $\frac{\Delta z}{r_p \Delta \varphi}$ the gradient at the contact point, μ_k the sliding friction between ball and profile, μ_s the sliding friction between the clutch ring and the housing and $\operatorname{sign}(\dot{\varphi})$ the direction of rotation of the respective characteristic curve. The spring force F_c is determined by the spring stiffness and the compression due to the displacement of the ball and the preload biasing distance.

Measurements of the clutch using a digital microscope with a 3D measurement function show that the profile is more rounded than the generic trapezoidal shape presented in [12].

The friction occurring in the components was summarized in two parameters. There is no exact assignment of the friction parameters to the friction contact. For the simulation, it is assumed that the friction conditions do not change significantly at different clutch stages and operating points, and therefore parameterization via a test is expedient. The friction behavior was determined on the basis of a measurement of the clutch on a test bench.

2.2 Implementation of the Clutch Model

The behavior of the torque-limiting clutch is embedded as a table of characteristic curves (as lookup-table) in simulation. These curves can be implemented as a Simulink block or as a Simscape component. The structure of the simulation is shown in Figure 4.

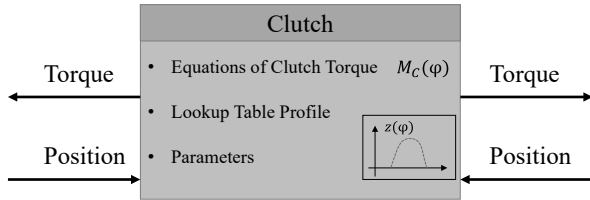


Figure 4: Implementation of the clutch in the simulation model.

Based on the determined relationships, the transmittable torque is converted into a characteristic curve as a function of the clutch position. These characteristic curves represent the torque of the clutch via the angular position of the clutch. Within the simulation, separate characteristic curves are used for the two directions of rotation of the torque-limiting clutch.

Other necessary properties for modeling the clutch are added in separate elements of the simulation. These include the inertia of the clutch ring and the planetary gear, as well as the backlash for the gear and clutch.

A pre-calculation of the characteristic curves is performed for the profile at hand before the simulation is carried out. First, the pre-calculation of the displacement of the ball, through the profile and the point where the profile and the ball contact, is performed. Using the formula, the maximum torque can be determined from the position of the ball.

These pre-calculations are prepared in characteristic curves as a lookup table for the simulation and thus allow a very performant calculation.

In this modeling, the two sliding frictions μ_s (clutch ring / housing) and μ_k (ball / clutch detent) are assumed. The sliding friction is assumed to be independent of the velocity additionally static friction effects were neglected here. Furthermore, small gear losses were considered in the modeling of the gearbox. Damping is also assumed within the planetary gear. It represents the contacts between the components and primarily serves to improve the simulation stability. The behavior of the clutch is shown in Figure 5.

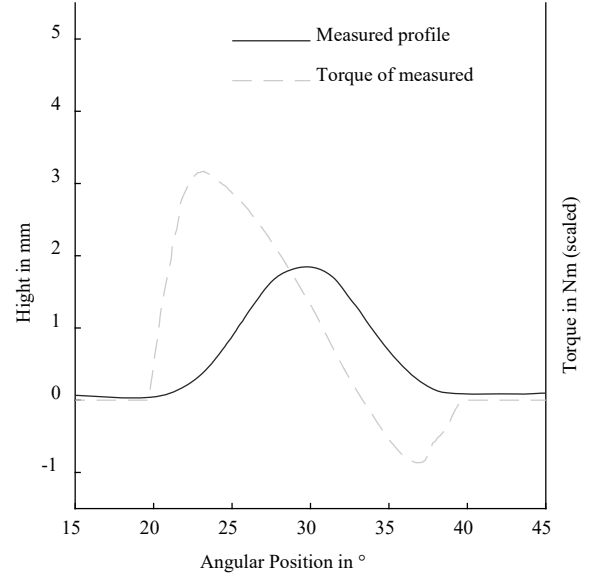


Figure 5: Measured profile of the clutch detents and torque of the clutch calculated via the simulation as a function of the angular position.

2.3 Clutch Behaviour in the Simulation

The simulation of the torque-limiting clutch allows the analysis and initial estimation of the influencing parameters on the clutch behavior (see Figure 5). The asymmetry of the torque-on-angle curve is remarkable. This can be explained by the friction and the backlash. The frictions act against the drive direction and are additively superimposed with the redirected spring force.

At the rising flank, both the redirected force and the friction force point in the opposite direction to the direction of motion, which increases the total amount of force. At the falling flank of the detent, the two forces point in opposite directions, reducing the overall applied force. At the top of the detent, the slope is very shallow, and the friction force is high, making the friction force dominant over the redirection force.

As a result, we see a flattening torque curve in the

area of the peak, instead of the zero crossing in a curve without friction. This zero crossing is shifted further into the range of the falling edge. The backlash (sometimes called lash, is caused by gaps between the parts, with result in lost motion) generates a negative torque pulse directly after the zero crossing (see Figure 6), since the direction of force changes there, the clutch ring is accelerated until it touches the other side of the backlash again.

As the moment of inertia of the clutch ring is very low, this occurs in a very short period of time and is therefore difficult to measure by torque measurement. In the simulation model, damping is applied to the backlash (with a translational hard stop). Since the profile continues to rotate in the backlash, the section of the profile is not visible in the torque curve.

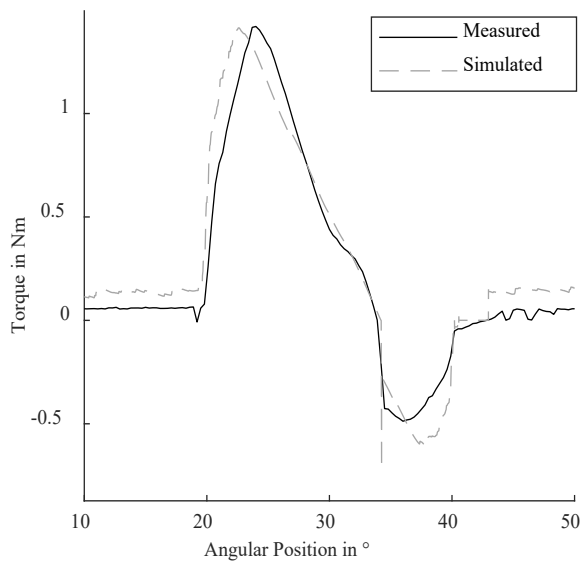


Figure 6: Torque of Simulation and Measurement on Ring Angle

2.4 Validation and Model Abilities

The simulation model replicates the relationship of the presented torque-limiting clutch between design parameters and functional behavior. A comparison between experiment and simulation is shown in Figure 7. The simulation calculates the transmittable clutch torque as a function of the angular position. The simulated clutch torque is compared with experiments on the test rig. The preload levels 5, 10 and 15 of the clutch were used for this purpose both on the test rig and in the simulation. Comparison of simulation with mechanical experiments shows good correlation for the present torque-limiting clutch. Figure 7 shows that the peak torque of the clutch can be predicted at different clutch levels with the simulation.

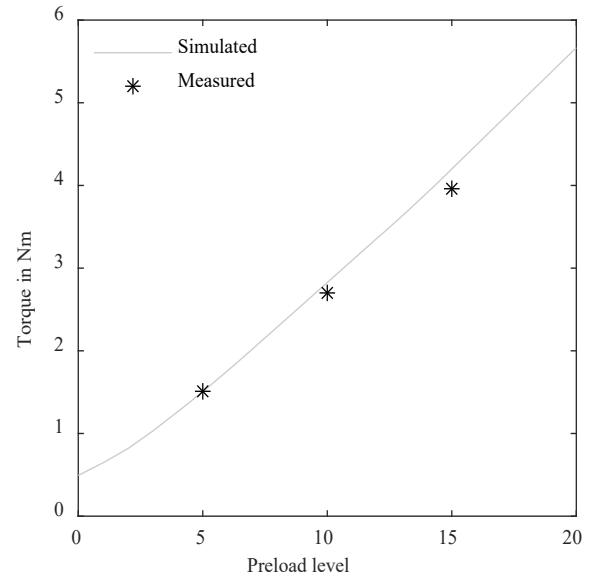


Figure 7: Relationship between preload level and peak torque of the clutch and simulation at different clutch levels

The simulation model allows the maximum clutch torques to be predicted as a function of the clutch level with a maximum deviation of 6%. The deviation is considered almost constant over the clutch stages. In the model, different design variants of the clutch can be simulated via different profile curves.

Customization. The model of the torque-limiting clutch nevertheless allows the customization of individual design parameters to be estimated by variation.

Different profile shapes can be considered in the simulation. When the geometry is changed, release behavior can be predicted. The behavior of an alternative clutch is shown in Figure 8. A comparison of the simulation results with generic descriptions [12], shows a high functional behavior of the simulation model. The simulation must be parameterized with multiple parameters. These parameters can be varied to estimate the behaviour of different clutches. Some of these parameters are simple values such as the profile radius, ball radius, preload distance, spring resistance, ball to profile friction and disk friction. The profile, on the other hand, is parameterized via the height as a function of the angle.

Profile Variation. Any reasonable profile design can be implemented. Figure 8 compares two different profile shapes with regard to their release behavior. Thus,

custom designs like trapezoids or notches and hight profiles from measurements can be used. Profiles with self-locking designs can create nonsensical simulation results. These designs would turn the physical clutch nonfunctional. Simulations are used to vary the profile shape for prediction to represent different behaviors.

The model is not validated for design changes in the clutch profiles. As long as there are no undercuts or longer vertical profile components in the geometry, a prediction of the behavior is possible.

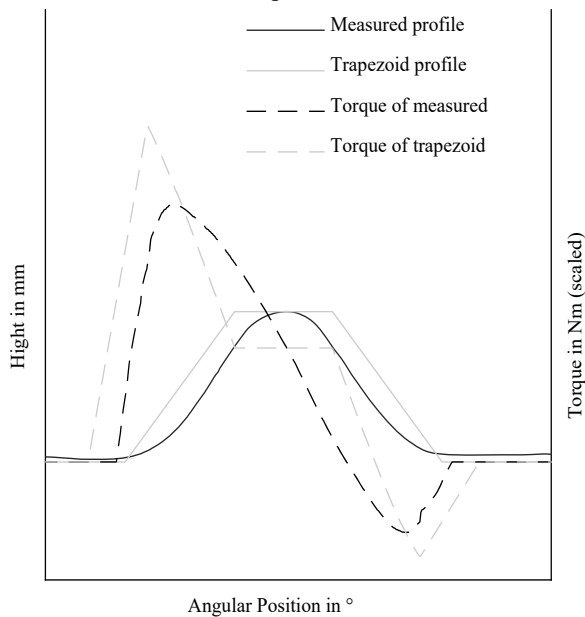


Figure 8: Relationship between the shape of the profile and the torque of the clutch when the geometry of the profile is customized.

3 Discussion

The simulation model is able to reproduce the behavior of the torque-limiting clutch. The model can be used for different clutches and adapted profile shapes to predict the behavior.

The pre-calculation of the ball center of gravity leads to a fast execution of the simulation. The representation in a Simulink model enables real-time execution. Thus, the simulation can also be used as a virtual subsystem in mixed physical virtual studies.

One approach to validate development variants on test benches and to integrate them into the overall system is the X-in-the-loop (abbr. XiL) approach [13]. The XiL approach supports this integration and enables the functional testing of a component or a subsystem (System-in-Development) in test benches by mapping the remaining

subsystems through suitable physical or virtual models [13] (as Connected Systems). Such test benches can be used to examine subsystems in a system context.

Limitations. The friction effects are implemented as simplified substitute effects, and do not fully represent the physical relationship. Furthermore, possible frictions are not considered at all surface pairs. The concrete behavior of the balls and the related friction behavior of the torque-limiting clutch can be modeled in more detail. It is still unclear how the balls behave in dynamic conditions. For example, the balls could lift off from the clutch ring. In this way, any interactions of the friction parameters with the functional behavior could be fully modeled and thus enable a more accurate prediction even in the case of changing material properties and the associated changes in the friction parameters. Static friction effects are also not part of the simulation model. Since the friction is not completely determined from the shape, a design of a clutch without a reference clutch, is not possible. If the parameters differ too much from the reference shape, the simulation model becomes increasingly invalid. A high quality of the analysis of the reference clutch is also necessary, since even small deviations in some parameters, such as the slope in the profile, have a large effect on the resulting torque.

4 Outlook

The tests have shown that the simulation model is suitable for mapping the relationship between the design parameters and the limited clutch torque.

If a higher reproduction quality is desired, further analysis and improvements to the models are necessary. Optimization is possible with regard to the behavior of the ball. Due to its real-time capability, the model is already suitable for use as a virtual subsystem for investigations in the context of XiL, for instance [14]. The use of the simulation model of the torque limiting clutch as part of a digital twin [15] can also be considered.

References

- [1] Matthiesen, S. *Gestaltung – Prozess und Methoden*. In: Bender, B.; Gericke, K. (Hrsg.). *Pahl/Beitz Konstruktionslehre*. Berlin, Heidelberg. Springer Berlin Heidelberg; 2021, S. 397–465.

- [2] Gold, PW., Schelenz, R, Koenen, H, Pflieger, E. *FVA Bericht : Abbildung und Simulation von mechanischen Ueberlastkupplungen*. Antriebstechnik; 2001, S. 58–61.
- [3] Gawas, NM, Patkar, MS, Gawade, PB. *Design and Analysis of Spring-Ball Clutch Torque Limiter*. International Journal of Engineering Development and Research; 2016.
- [4] MathWorks Deutschland. *Double-sided, spring-loaded translational detent - MATLAB*. Online verfügbar unter <https://de.mathworks.com/help/physmod/sdl/ref/translationaldetent.html>, zuletzt geprüft am 07.01.2021.
- [5] MapleSim. *Translational Detent - MapleSim Help*. Online verfügbar unter <https://www.maplesoft.com/support/help/MapleSim/view.aspx?path=DrivelineComponentLibrary/translationalDetent>, zuletzt geprüft am 07.01.2021.
- [6] Dennig, H-J. *Entwicklung einer schnell schaltenden Bremse und Kupplung für Linearbewegungen zum Überlastschutz in Werkzeugmaschinen*. Universität Stuttgart; 2009.
- [7] Paland, D. *Entwicklung eines Simulationsmodells einer Sperrkörperkupplung*. Karlsruhe: Karlsruher Institut für Technologie (KIT), IPEK - Institut für Produktentwicklung, Unveröffentlichte Bachelorarbeit; 2019.
- [8] Schmitt, TL., Andres, M. *Methoden zur Modellbildung und Simulation mechatronischer Systeme*. Wiesbaden: Springer Fachmedien Wiesbaden; 2019.
- [9] Grauberger, P, Bremer, F, Sturm, C, Hoelz, K, Wessels, H, Gwosch, T, Wagner, R, Lanza, G, Albers, A, Matthiesen, S. *Qualitative Modelling in Embodiment Design - Investigating the Contact and Channel Approach Through Analysis of Projects*. Proceedings of the Design Society: DESIGN Conference; 2020, S. 897–906.
- [10] Matthiesen, S, Grauberger, P, Bremer, F, Nowoseltschenko, K. *Product models in embodiment design: an investigation of challenges and opportunities*. SN Applied Sciences; 2019, S. 423.
- [11] Gwosch, T. *Antriebsstrangprüfstände zur Ableitung von Konstruktionszielgrößen in der Produktentwicklung handgehaltener Power-Tools*. In: Albers, A.; Matthiesen, S. (Hrsg.). *Forschungsberichte Band 117*. Karlsruhe. Karlsruher Institut für Technologie (KIT); 2019.
- [12] Schweizer, HH. *1500 Fragen und Antworten: Elektrowerkzeuge und ihre Anwendung*. [Anwendungen, Anwendungstabellen, Auswahldiagramme]. Konstanz: Paul Christiani, Technisches Institut für Aus- und Weiterbildung GmbH & Co.KG; 2003.
- [13] Albers, A, Behrendt, M, Klingler, S, Matros, K. *Verifikation und Validierung im Produktentstehungsprozess*. In: Lindemann, U. (Hrsg.). *Handbuch Produktentwicklung*. München. Hanser; 2016, S. 541–569.
- [14] Steck, M, Gwosch, T, Matthiesen, S. *Scaling of Rotational Quantities for Simultaneous Testing of Powertrain Subsystems with Different Scaling on a X-in-the-Loop Test Bench*. Mechatronics; 2020, S. 102425.
- [15] Stark, R, Anderl, R, Thoben, K-D, Wartack, S. *WiGeP-Positionspapier: „Digitaler Zwilling“*. ZWF Zeitschrift für wirtschaftlichen Fabrikbetrieb; 2020, special, S. 47–50.

Modellierung des Betriebsverhaltens eines Diesel-Industriemotors als Stromerzeuger

Daniel Jörss¹, Bert Henrik Herrmann¹, Christian Fink^{1*}

¹. Fakultät für Maschinenbau, Hochschule Wismar, Philipp-Müller-Str. 14, 23966 Wismar, Germany

*Christian.Fink@hs-wismar.de

Abstract. Im Rahmen des laufenden BMBF-finanzierten und durch den Projektpartner Caterpillar unterstützten Forschungsprojektes SIDYN soll das Betriebsverhalten eines Stromerzeugers in Inselnetzen mit dynamischen Netzlasten simulativ beschrieben werden. Der hier präsentierte aktuelle Entwicklungsstand beschreibt das thermodynamisch-mechanische Motormodell, an welches im weiteren Projektverlauf das mechanisch-elektrische Generator-Netz-Modell anschließen soll. Zur Verifizierung eines Simulationsansatzes für den Industriemotor wurden Messungen an einem Laboraufbau an der Hochschule Wismar durchgeführt. Ziel dabei ist es, eine klare Schnittstellenbeschreibung zu entwickeln, die es ermöglicht, verschiedene Lastszenarien und deren Rückwirkung auf das Systemverhalten zu simulieren. Dabei wird ein in [2] beschriebener Ansatz verfolgt und in MATLAB® Simulink® umgesetzt. Zur Darstellung des Regelverhaltens seitens des Industriemotors wurde eine Schnittstellenbeschreibung der zeitlich variablen Momente und Winkelgeschwindigkeiten am Antriebsstrang betrachtet. Bezugnehmend auf den vorliegenden Versuchsaufbau wurden in erster Näherung gute Ergebnisse mit dem vorliegenden Modell erreicht. So zeigen die berechneten und simulierten Zylinderdruckverläufe in verschiedenen stationären Betriebspunkten eine hohe Übereinstimmung. Die im weiteren Projektverlauf angestrebte Berechnung dynamischer Betriebszustände erfordert jedoch verschiedene Erweiterungen des Systemmodells sowie auch des Motorprüfstandes.

Einleitung

Zur Erzeugung elektrischer Energie in Inselnetzen werden derzeit im maritimen sowie im stationären Inselbetrieb vorrangig GENSETS (Set aus Generator und Verbrennungsmotor) eingesetzt. Hierbei wird die hohe Energiedichte fossiler Brennstoffe durch die Kopplung eines Stromgenerators mit einem Verbrennungsmotor (meist Diesel- oder Gasmotoren) in elektrische Energie umgewandelt. Bereits im Auslegungsprozess der Anlage ist es zunehmend gefordert, Wechselwirkungen zwischen den

Systemkomponenten zu quantifizieren und eine optimierte Systemkonfiguration und Betriebsweise zu identifizieren. Auswirkungen dynamischer Lastwechsel im elektrischen Netz und die Reaktion des Verbrennungsmotors haben in Inselnetzen beispielsweise direkte Auswirkungen auf Schwankungen der Netzfrequenz. Insbesondere bei Anwendungen mit hohen elektrischen Lasten im Inselbetrieb, wie z.B. bei Zement- und Stahlwerken oder auch Schiffen mit Hybridantriebssystemen können erhebliche Einflüsse auf die Stabilität der Netzfrequenz entstehen und negative Auswirkungen auf andere Netzkomponenten hervorrufen. Für die Auslegung und Betrieb entsprechender Inselnetze ist es von großer Bedeutung, die zu erwartenden Wechselwirkungen zuverlässig quantitativ bestimmen zu können. Bislang werden solche Berechnungen aufwändig von Hand oder in Teilsimulationen durchgeführt. Unsicherheiten führen dabei oftmals zur Wahl einer überdimensionierten Anlagenkategorie mit entsprechend hohen Beschaffungskosten.

Grundsätzlich werden herkömmliche Simulationen solcher Systeme auf Basis von drehzahlbasierten Modellen berechnet [7]. Der Ansatz des Projektes beinhaltet jedoch die Annahme, dass durch die Entwicklung eines Modells auf Basis drehwinkel aufgelöster Prozessrechnung eine präzisere Aussage zum Systemverhalten getätigt werden kann und somit die Anlagenkonfiguration hinsichtlich deutlich mehr Parametern optimierbar ist. Als Ergebnis des Auslegungsprozesses kann ggf. eine wirtschaftlichere Anlage gewählt werden, die dennoch eine stabile Systemleistung mit geringen Netzschwankungen ermöglicht.

Neben einer sicheren und bestmöglichen Systemleistung können auch weitere ökonomische und ökologische Gesichtspunkte, wie Effizienz und Emissionen, als Zielgrößen einer modellbasierten Systemauslegung definiert werden.

Bei der Simulation sollen zwei Systemmodelle an einer Schnittstelle zusammengeführt werden. Dabei

handelt es sich um ein mechanisch-thermodynamisches Motormodell sowie ein mechanisch-elektrisches Generator-Netz-Modell. Zur Modellentwicklung sollen zunächst ein 2,9 l Turbodiesel- Industriemotor sowie ein Generator als Asynchronmaschine abgebildet werden, wie sie im Motorenlabor der Hochschule Wismar existieren. Die Schnittstelle wird über einen zeitlich und drehwinkelaufgelösten Drehmomentverlauf realisiert, um hochaufgelöst Lastwechselspiele zwischen den Maschinen, inklusive der gedämpften, elastischen Kupplung, beschreiben zu können.

In diesem Dokument soll für das Teilmodell des Verbrennungsmotors der bis dato entwickelte Simulationsansatz beschrieben, sowie erlangte Simulationsergebnisse mit Messergebnissen des Motorprüfstandes verglichen werden.

1 Die Simulationsumgebung

Zur Simulation des Dieselmotors wurde ein Berechnungsansatz auf Grundlage einschlägiger 0D-Modelle sowie verfügbarer Motordaten gewählt.

Obwohl eine Reihe kommerzieller Tools zur Berechnung von Antriebssystemen verfügbar ist, wurde sich unter Berücksichtigung der Projektziele für die Eigenentwicklung eines entsprechenden Simulationsmodells entschieden. Gründe hierfür sind:

- Genaue Kenntnis des Aufbaus der Modelle
- Spezielle Abstimmung der Modelle auf Gegebenheiten von Großmotoren und Stromerzeugern in Einzel- und Mehrmotorenanlagen mit gegenseitiger Wechselwirkung
- Freie Wahl des Detaillierungsgrades und der Anwenderkomplexität der Teilmodelle und des Gesamtsystems
- Ausbaufähigkeit des Modells hinsichtlich weiterer Schnittstellen, wie z.B. Ankopplung an Wärmenetze bei Abwärmenutzung

Die Wahl der Simulationsumgebung fällt auf das Programm MATLAB® Simulink®. Hierin ist eine flexible Struktur abbildbar und es können sowohl existierende Module als auch eigener Code genutzt werden. So können physikalische Zusammenhänge nachvollziehbar und adaptierbar programmiert werden.

2 Modellbeschreibung

Zur Simulation des Verbrennungsmotors wird ein nulldimensionales Modell verwendet, welches auf dem 1. Hauptsatz der Thermodynamik beruht, in dem der Brennraum ein instationäres, offenes System darstellt [1].

Die Energiebilanz [1], abgeleitet nach dem Kurbelwinkel φ , wird im Allgemeinen wie folgt beschrieben:

$$\frac{dU}{d\varphi} = -p \frac{dV}{d\varphi} + \frac{dQ_B}{d\varphi} - \frac{dQ_W}{d\varphi} + h_E \frac{dm_E}{d\varphi} - h_A \frac{dm_A}{d\varphi} \quad (1)$$

Für die Lösung der Bilanzgleichung muss in der Simulation eine Variable vorgegeben werden, wie z.B. der Zylinderdruck aus Messdaten oder ein empirisch ermittelter Brennverlauf (Wärmezufuhr aufgrund Verbrennung). Die innere Energie im Brennraum wird über die kalorische Zustandsgleichung für ideale Gase berechnet.

$$dU = m \cdot c_v \cdot dT \quad (2)$$

In Anlehnung an [2] wird in der Simulation die Gleichung 2 nach der Zeit abgeleitet, so verschwindet die Ableitung der konstanten Bezugstemperatur und man erhält

$$\Delta \dot{U} = m \cdot c_v \cdot \dot{T} \quad (3)$$

Mit diesem Ansatz [2] lässt sich der 1. Hauptsatz als Differentialgleichung 1. Ordnung wie folgt formulieren

$$\dot{T} = \frac{-p \cdot \frac{dV(\varphi)}{dt} + \dot{Q}_B - \dot{Q}_W + \dot{H}_E - \dot{H}_A}{m \cdot c_v} \quad (4)$$

Für die spezifische Wärmekapazität c_v sind in der Simulation Stoffdaten [3] in Polynomgleichungen, in Abhängigkeit der berechneten Temperatur, für verschiedene Luftverhältnisse hinterlegt, um damit höhere Genauigkeiten zu erzielen.

Die Berechnung des Zylinderdruckes lässt sich mit Hilfe der idealen Gasgleichung berechnen.

$$p = \frac{m \cdot R \cdot T}{V} \quad (5)$$

Neben der Bestimmung von Druck und Temperatur ist es noch erforderlich, den zugeführte Wärmestrom \dot{Q}_B aus dem Brennstoff zu berechnen. Die Energieumsetzung während der Verbrennung wird mittels eines Vibe-

Ersatzbrennverlaufs dargestellt [8][4]. Um den dieselmotorischen Brennverlauf möglichst realitätsnah darzustellen, wird für die Simulation eine Überlagerung aus drei Vibe-Funktionen verwendet. Diese sollen die drei typischen Phasen vorgemischte Verbrennung, Hauptverbrennung und Nachverbrennung beschreiben [4]. In Abbildung 1 sind die einzelnen Brennverlaufs-funktionen sowie deren Überlagerung exemplarisch dargestellt.

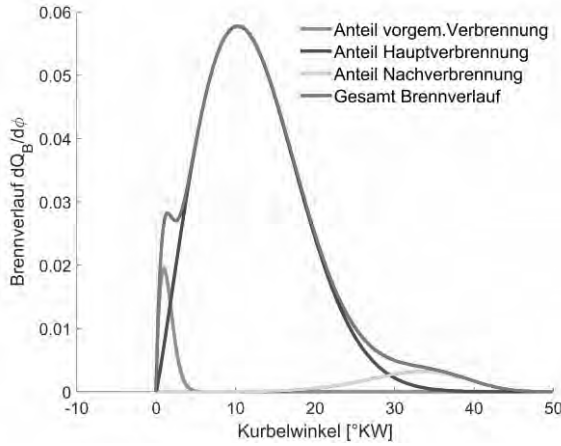


Abbildung 1: Modellierung Brennverlauf

Der Brennverlauf wird im Vorfeld über ein Programm in MATLAB® erzeugt und in Simulink® an ein 1D Lookup-Table als Durchbrennfunktion weitergegeben. Mit Hilfe des Newton'schen Ansatzes lässt sich die abgeführte Wärmeleistung \dot{Q}_W , durch Wärmeübergang an Zylinderwand und Kolben, beschreiben.

$$\dot{Q}_W = \alpha \cdot A(\varphi) \cdot (T_{Gas} - T_{Wand}) \quad (6)$$

In der Gleichung 6 ist A die Wärme übertragende Fläche, die sich aus Zylinderkopffläche, Kolbenbodenfläche und der momentanen Mantelfläche des Brennraumes je nach Kolbenposition zusammensetzt.

Für den Wärmeübergangskoeffizienten α wurde sich in der Simulationsumgebung für die Gleichung 7 nach Woschni-Huber [3] entschieden, da diese neben dem Hochdruckteil (Kompression und Expansion) noch den Ladungswechsel (Ausschieben und Ansaugen) mit berücksichtigt.

$$\alpha = 130 \cdot d^{-0,2} \cdot p^{0,8} \cdot T^{-0,53} \cdot (C_1 \cdot v)^{0,8} \quad (7)$$

Die dimensionslose Konstante C_1 ist mit folgenden Werten angegeben:

$$C_1 = 2,28 + 0,308 \frac{v_u}{v_{KM}} \text{ für den Hochdruckteil}$$

$$C_1 = 6,18 + 0,417 \frac{v_u}{v_{KM}} \text{ für den Ladungswechsel}$$

Der Geschwindigkeitsterm v wird in der Gleichung 8 nach [3] wie folgt beschrieben:

$$v = v_{KM} \left[1 + 2 \cdot \left(\frac{V_c}{V} \right)^2 \cdot p_i^{-0,2} \right] \quad (8)$$

Dabei ist p_i der indizierte Mitteldruck, der in der Simulationsumgebung als integrale Größe aus der Volumenänderungsarbeit, bezogen auf das Hubvolumen des Zylinders, berechnet wird.

Die Terme \dot{H}_E und \dot{H}_A sind die Enthalpieströme der durch die Ventile ein- und ausströmenden Gasmassen, die in der Simulation wie folgt formuliert werden [4].

$$\dot{H}_E = \dot{m}_E \cdot (u_E + R_E \cdot T_E) \quad (9)$$

$$\dot{H}_A = \dot{m}_A \cdot (u_A + R_A \cdot T_A) \quad (10)$$

Der Index E kennzeichnet den Zustand knapp vor dem Einlassventil und in analoger Weise der Index A den Zustand nach dem Auslassventil. Die Berechnung der spezifischen inneren Energie beruht auf Stoffdaten aus [3], die in Polynomgleichungen, als Funktion der Temperatur, für das jeweilige Luftverhältnis überführt wurden. Zur Berechnung des Massenstroms durch die Ventile im Ladungswechsel wird folgende Gleichung 11 bzw. 12 verwendet [3].

$$\dot{m}_E = \mu \cdot A_{V E(\varphi)} \cdot \frac{p_E}{\sqrt{R \cdot T_E}} \cdot \frac{2\kappa}{\kappa - 1} \cdot \left[\left(\frac{p_Z}{p_E} \right)^{\frac{2}{\kappa}} - \left(\frac{p_Z}{p_E} \right)^{\frac{(\kappa+1)}{\kappa}} \right] \quad (11)$$

$$\dot{m}_A = \mu \cdot A_{V A(\varphi)} \cdot \frac{p_Z}{\sqrt{R \cdot T_Z}} \cdot \frac{2\kappa}{\kappa - 1} \cdot \left[\left(\frac{p_A}{p_Z} \right)^{\frac{2}{\kappa}} - \left(\frac{p_A}{p_Z} \right)^{\frac{(\kappa+1)}{\kappa}} \right] \quad (12)$$

Entsprechend stehen die Indizes Z für Zylinder, E für Einlass und A für Auslass, jeweils betrachtet knapp vor beziehungsweise nach dem Ventil.

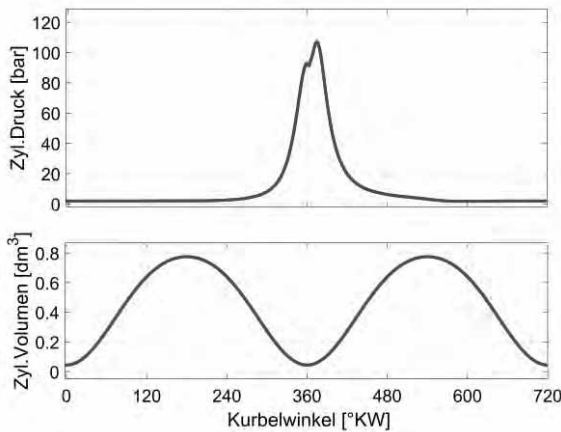


Abbildung 4: oben: Zylinderdruckverlauf über 720°KW
unten: Zylindervolumen über 720°KW

Abbildung 4 zeigt den simulierten Zylinderdruckverlauf sowie das veränderliche Zylindervolumen über ein Arbeitsspiel (720° KW). Mit der Aufwärtsbewegung des Kolbens verkleinert sich das Zylindervolumen und es beginnt die Kompressionsphase, wodurch Druck und Temperatur ansteigen. Bei ca. 360° KW setzt die Verbrennung ein und der Druck steigt an. Mit der Abwärtsbewegung des Kolbens vergrößert sich das Zylindervolumen und es folgt die Expansionsphase, in der Druck und Temperatur im weiteren Verlauf sinken. Kurz vor 480° KW öffnet das Auslassventil und es beginnt das Auschieben der Abgase. Für die Modellierung eines 4-Zylindermotors wird die Energiebilanz für jeden Brennraum einzeln berechnet. Dabei muss bei der Integration des Drehwinkels aus der Winkelgeschwindigkeit eine Verschiebung berücksichtigt werden, die beim 4-Zylindermotor dem Zündabstand von 180° KW entspricht.

Für die Berechnung des Drehzahlverlaufs werden jeweils die Terme der am Kolben verrichteten Volumenänderungsarbeit $-p \frac{dv}{d\varphi}$, abgeführte Nutzleistung, notwendige Reibleistung sowie Leistungen zur oszillierenden Beschleunigung der Kolben herangezogen. Die Ermittlung des Reibmitteldrucks erfolgt dabei durch einen Polynomansatz von Chen & Flynn [5], welcher eine gute Adaptierbarkeit an die Rand- und Betriebsbedingungen erlaubt. In der folgenden Abbildung 5 ist eine schematische Darstellung zur Berechnung der Drehungleichförmigkeit in Simulink® dargestellt.

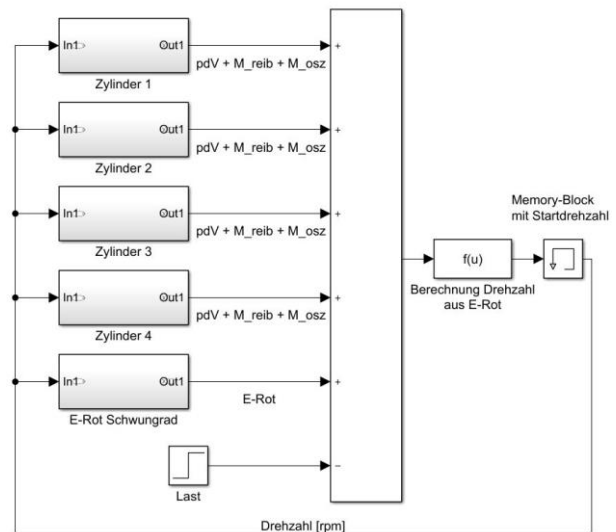


Abbildung 5: schematische Darstellung

Über den „Memory-Block“ muss im Vorfeld eine Startdrehzahl definiert werden, um die Energiebilanz berechnen zu können. In den Subsystemen „Zylinder 1“ bis „Zylinder 4“ wird jedes Einzelmoment, unter Berücksichtigung des oszillierenden Massenmomentes und des Reibmomentes, berechnet. Die Lastanforderung in diesem System ist als Konstante hinterlegt, welche sich je nach Simulationsbedingungen erhöhen oder verringern lässt. Der Energiezustand des Schwungrades lässt sich mit Gleichung 13 beschreiben.

$$E_{Rot} = \frac{1}{2} \cdot J \cdot \omega^2 \quad (13)$$

In Abbildung 6 ist ein simulierter Momentenverlauf für einen Zylinder, über ein Arbeitsspiel, dargestellt.

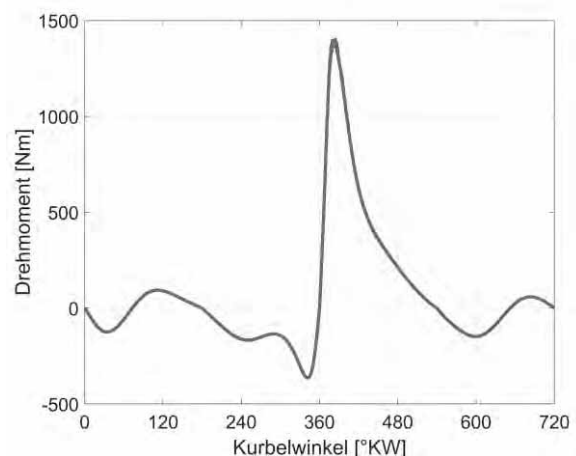


Abbildung 6: simulierter Momentenverlauf

An der Summenstelle werden alle einfließenden Energien zusammengeführt. Durch Umstellen der Gleichung 13 lässt sich der Drehzahlverlauf wie folgt berechnen

$$n = \sqrt{\frac{E_{Rot\ neu}}{2 \cdot J \cdot \pi^2}} \quad (14)$$

In Abbildung 7 ist die simulierte Ungleichförmigkeit im Drehzahlverlauf für ein Arbeitsspiel nach Gleichung 14 über der Zeit, bei einer Nenndrehzahl von 1500 min^{-1} , dargestellt.

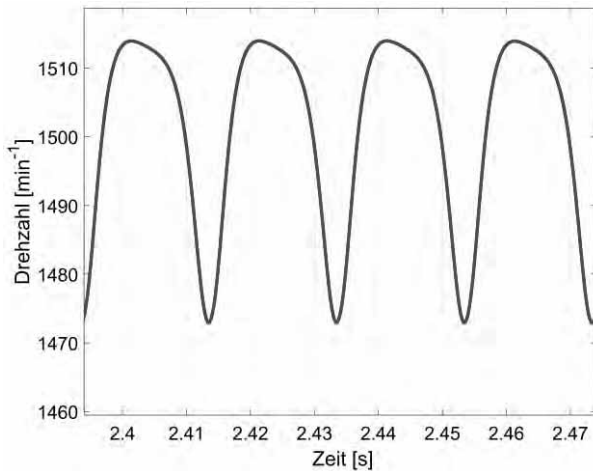


Abbildung 7: simulierte Drehungleichförmigkeit

Da die Drehungleichförmigkeit des Motors einen erheblichen Einfluss auf eine mit ihm gekoppelte E-Maschine hat, muss zusätzlich das Verhalten eines elastischen Kupplungselementes simuliert werden. Nach dem dynamischen Grundgesetz für die Drehbewegung ergeben sich für die Drehmassen folgende Gleichungen [6].

$$-k_D(\varphi_M - \varphi_G) - d_D(\dot{\varphi}_M - \dot{\varphi}_G) + M = J_M \cdot \ddot{\varphi}_M \quad (15)$$

$$k_D(\varphi_M - \varphi_G) + d_D(\dot{\varphi}_M - \dot{\varphi}_G) = J_G \cdot \ddot{\varphi}_G \quad (16)$$

In den Gleichungen 15 und 16 ist $d_D(\dot{\varphi}_M - \dot{\varphi}_G)$ das Dämpfungsmoment und $k_D(\varphi_M - \varphi_G)$ das Moment aus der Verdrehung beider Drehmassen zueinander. M ist das Erregermoment, das von der Motorseite eingeleitet wird.

Nach der höchsten Ableitung umgestellt, ergibt sich folgende Gleichung 17

$$-k_D \left(\frac{1}{J_M} + \frac{1}{J_G} \right) \cdot (\varphi_M - \varphi_G) - d_D \left(\frac{1}{J_M} + \frac{1}{J_G} \right) \cdot (\dot{\varphi}_M - \dot{\varphi}_G) + \frac{M}{J_M} = \ddot{\varphi}_M - \ddot{\varphi}_G \quad (17)$$

In Abbildung 8 ist ein simulierter Drehzahlverlauf vor und hinter dem elastischen Kupplungselement dargestellt. Es ist zu erkennen, dass sich die Drehungleichförmigkeit des Verbrennungsmotors durch die elastische Kupplung, deutlich reduziert, auf den Generator überträgt.

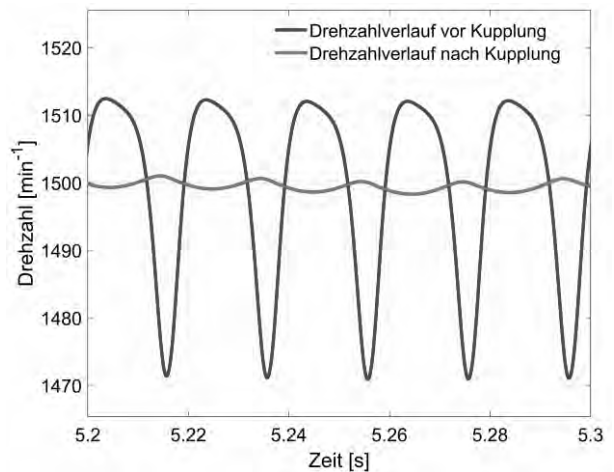


Abbildung 8: simulierte Drehungleichförmigkeit inklusive Kupplung

3 Modellvalidierung

Zur Verifizierung des Simulationsmodells, werden die Berechnungsergebnisse mit Messdaten des Versuchsmotors aus dem Motorenlabor der Hochschule Wismar verglichen.

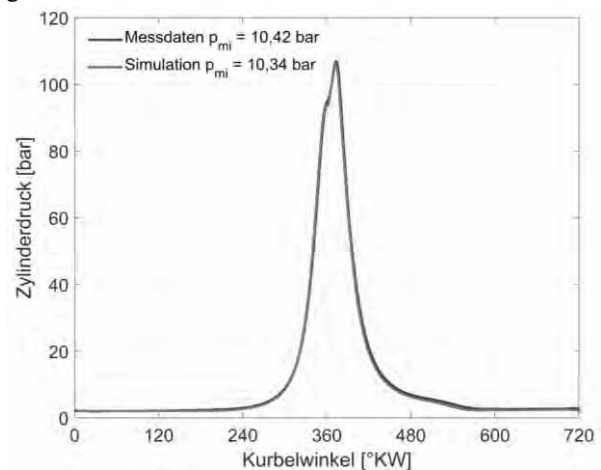


Abbildung 9: Zylinderdruckverlauf eines Zylinders über ein Arbeitsspiel im Vergleich von Messung zu Simulation

In Abbildung 9 ist ein Vergleich des Zylinderdruckverlaufes über einen komplettes Arbeitsspiel bei einer Motordrehzahl von 2000 min^{-1} und einem Drehmoment von 200 Nm dargestellt. Als Eingangsparameter für die Simulation wurde ein aus den Messungen ermittelter Brennverlauf eingesetzt. Ein Vergleich der ermittelten Zylinderdruckverläufe zeigt, dass Simulation und Berechnung in hohem Maße übereinstimmen. Dies ist eine wichtige Voraussetzung für eine genaue Beschreibung des zeitlichen Drehmomentverlaufes an der Schnittstelle zwischen Verbrennungsmotor und angeschlossenen Generator. Ein Vergleich des Mitteldruckes als integrale Größe zur Charakterisierung der Motorlast ergibt eine Abweichung von kleiner 1 % zwischen Simulation und Messung. Ähnlich gute Ergebnisse wurden auch in anderen Betriebspunkten festgestellt, was zumindest für die stationären Betriebspunkte die Anwendbarkeit des Simulationsansatzes und der gewählten Modelle bestätigt.

4 Zusammenfassung und Ausblick

Im Rahmen des BMBF-finanzierten Forschungsvorhabens SIDYN soll das Betriebsverhalten eines Stromerzeugers in Inselnetzen mit dynamischen Netzlasten auf Grundlage eines modularen Simulationsmodells abgebildet werden. Das Modell soll die Möglichkeit zur Variation unterschiedlicher Motor- und Netzkonfigurationen sowie verschiedener Lastszenarien bieten.

Basierend auf thermodynamisch-mechanischen Zusammenhängen sowie einzelnen, speziell für Verbrennungsmotoren entwickelten Teilmodellen wurde dafür das hier vorgestellte Gesamtmodell eines 4-Zylinder 4-Takt Industrie-Dieselmotors in Simulink® aufgebaut. Ein Vergleich der mittels des Modells berechneten Ergebnisse zeigt dabei in stationären Betriebspunkten sehr gute Übereinstimmungen mit Messergebnissen von einem Motorprüfstand, was die Anwendbarkeit des gewählten Simulationsansatzes unterstützt.

Da die Abbildung hochdynamischer Betriebszustände ein wesentliches Ziel des Vorhabens ist, müssen im weiteren Projektverlauf sowohl das Motormodell als auch der Motorprüfstand entsprechend weiterentwickelt werden. Im konkreten Fall bedeutet dies, die Modellierung und messtechnische Erfassung des dynamischen Verhaltens des Abgasturboladers. Es ist darüber hinaus zu prüfen, welchen Einfluss Wärmeübergangsprozesse in Ansaug- und Abgasstrecke sowie eine Abgasrückführung auf das dynamische Betriebsverhalten haben.

Danksagung

Die hier vorgestellten Ergebnisse wurden im Rahmen des öffentlich finanzierten Projektes SIDYN (FKZ: 13FH043PX8) erarbeitet. Die Autoren möchten sich sowohl beim BMBF sowie auch bei dem Projektpartner Caterpillar Motoren GmbH & Co. KG für die Unterstützung des Vorhabens bedanken.

Nomenklatur

U	- innere Energie [J]
p	- Druck [Pa]
\dot{Q}_B	- Energiefreisetzung aus Brennstoff [J/s]
\dot{Q}_W	- Wandwärmestrom [W]
φ	- Kurbelwinkel [°]
KW	- Kurbelwinkel
h	- spezifische Enthalpie [J/kg]
m	- Masse [kg]
T	- Temperatur [K]
V	- Volumen [m ³]
c_v	- spezifische Wärmekapazität [J/(kg K)]
$\dot{H}_{E,A}$	- Enthalpiestrom [J/s]
\dot{m}	- Massenstrom [kg/s]
R	- Gaskonstante [J/(kg K)]
α	- Wärmeübergangskoeffizient [W/(m ² K)]
A	- wärmeübertragende Fläche [m ²]
d	- Bohrungsdurchmesser [m]
v_u	- Drallgeschwindigkeit [m/s]
v_{KM}	- mittlere Kolbengeschwindigkeit [m/s]
u	- spezifische innere Energie [J/kg]
p_i	- indizierter Mitteldruck [bar]
μ	- Durchflusskoeffizient [-]
A_v	- Schlitzzfläche [m ²]
κ	- Isentropenexponent [-]
J	- Trägheitsmoment [kg m ²]
ω	- Winkelgeschwindigkeit [s ⁻¹]
J_M	- Trägheitsmoment Motor [kg m ²]
J_G	- Trägheitsmoment Generator [kg m ²]
k_D	- Drehfedersteifigkeit [Nm/rad]
d_D	- Dämpfungskonstante [Nms/rad]
φ_M	- Verdrehwinkel Motor seitig [rad]
φ_G	- Verdrehwinkel Generator seitig [rad]
$\dot{\varphi}_M$	- Winkelgeschwindigkeit Motor seitig [rad/s]
$\dot{\varphi}_G$	- Winkelgeschwindigkeit Generator seitig [rad/s]
$\ddot{\varphi}_M$	- Winkelbeschleunigung Motor seitig [rad/s ²]
$\ddot{\varphi}_G$	- Winkelbeschleunigung Generator seitig [rad/s ²]

References

- [1] Maurer, T. *Einführung in die Realprozessrechnung von Verbrennungsmotoren*, Berlin: Springer-Verlag GmbH, 2020.
- [2] Scherf, H. *Modellbildung und Simulation dynamischer Systeme*, München: Oldenbourg Wissenschaftsverlag, 2010.
- [3] Pischinger, R., Klell, M., Sams, T. *Thermodynamik der Verbrennungskraftmaschine*, 3.Aufl., Wien, New York: Springer-Verlag, 2009.
- [4] Merker, G., Schwarz, C. *Grundlagen Verbrennungsmotor*, 4. Auflage, Wiesbaden: Vieweg + Teuber, 2009.
- [5] Chen, S., Flynn, P. „*Development of a Single Cylinder Compression Ignition Research Engine*“ in SAE Technical Paper 650733, 1965.
- [6] Jäger, H., Mastel, R., Knaebel, M. *Technische Schwingungslehre*, 8.Aufl., Wiesbaden: Springer Vieweg, 2013
- [7] Mrugowsky, H. *Drehstrommaschinen im Inselbetrieb*, 2.Aufl., Wiesbaden: Springer Vieweg, 2015.
- [8] Vibe, I. *Brennverlauf und Kreisprozess von Verbrennungsmotoren*, VEB Verlag Technik, Berlin, 1970

Simulation von Energiekonzepten in mittelständischen Gewerbegebieten

Maximilian Stange^{1*}, Tom Wolf¹, Benjamin Jacobsen²

¹ Fraunhofer-Institut für Werkzeugmaschinen und Umformtechnik IWU, Reichenhainer Straße 88, 09126 Chemnitz; *maximilian.stange@iwu.fraunhofer.de

² Professur für Energie- und Hochspannungstechnik, Technische Universität Chemnitz, Reichenhainer Straße 70, 09126 Chemnitz, Deutschland

Abstract. Dieser Beitrag beschreibt ein Vorgehen, um verschiedene Energiekonzepte in mittelständischen Gewerbegebieten zu generieren und diese mithilfe von Simulationen zu validieren und zu bewerten. Dabei liegt der Schwerpunkt auf der Kooperation der ansässigen Unternehmen.

den Bedürfnissen und Anforderungen der Nutzer und Stakeholder eines Systems Produkt- und Prozessbeschreibungen abgeleitet werden [2]. Der für diesen Beitrag relevante Teil des System Engineering Prozesses ist in Abbildung 1 dargestellt.

Einleitung

In Industrie- und Gewerbegebieten findet ein großer Teil der Wertschöpfung, der deutschen Wirtschaft statt. Damit einher geht auch ein signifikanter Energiebedarf, der heute noch zu einem erheblichen Teil durch die Nutzung fossiler Energieträger gedeckt wird. Viele Unternehmen in Gewerbegebieten sind bestrebt ihre eigene Energieversorgung zu optimieren. Demgegenüber fehlt in Gewerbegebieten häufig ein aktives Management, anders als in Industriegebieten oder speziell in Chemieparks, in denen Betreibergesellschaften Infrastruktur und Dienstleistung aus einer Hand zur Verfügung stellen. Dadurch zeichnen sich Gewerbegebiete häufig durch eine fehlende Kooperation, speziell im Bereich der Energieversorgung und -verteilung, der dort ansässigen Firmen aus.

Dabei stellen Gewerbegebiete einen Großteil der Gewerbeflächen dar. Das konkret untersuchte Gewerbegebiet liegt in der Stadt Limbach-Oberfrohna im Landkreis Zwickau, dort liegt der Anteil an Gewerbegebieten bei 74 % der gesamten Gewerbefläche [1].

Diese Herausforderung wird im Projekt GRIDS des Fraunhofer IWU und der TU Chemnitz aufgegriffen. In dem Projekt wird untersucht, wie eine Zusammenarbeit bei der Energieversorgung zu realisieren ist.

1 Ableitung der Energiekonzepte

Zur Entwicklung eines neuen Ansatzes von Energiekonzepten in mittelständischen Gewerbegebieten, wird der Systems Engineering Ansatz genutzt. Dabei handelt es sich um einen iterativen Prozess, indem aus

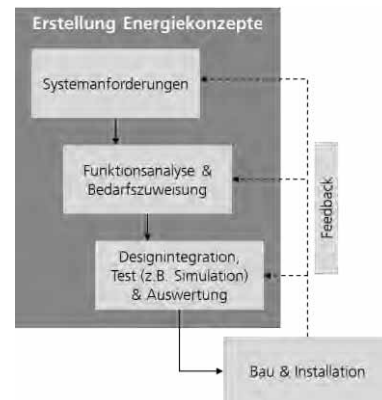


Abbildung 1: Vorgehensweise zur Erstellung der Energiekonzepte in Anlehnung an [2]

Im Zuge von GRIDS wurde das Gewerbegebiet Süd (kurz GG-Süd) in Limbach-Oberfrohna untersucht. Besonders wichtig ist bei der Aufnahme der Systemanforderungen die Identifikation der Nutzeranforderungen, die einen Teil der Systemanforderungen darstellen. Die zukünftigen Nutzer des zu definierenden Energiekonzeptes, sind die am Standort tätigen Unternehmen. Zielbild ist die Integration aller Nutzer des GG-Süd in ein gemeinsam genutztes Energieversorgungssystem. Die Gruppe der Nutzer ist als heterogen zu charakterisieren. Dies zeigt sich beispielsweise am Branchenmix der ansässigen Firmen. 60 % der Firmen können dem verarbeitenden Gewerbe zugeordnet werden, unterscheiden sich aber erheblich in ihrer Größe (sowohl Fläche, wie auch Mitarbeiteranzahl und Jahresumsatz) und den hergestellten Produkten. Der Rest teilt sich auf die Branchen Logistik sowie Handel auf. Die generellen Anforderungen der Nutzer an die Energieversorgung

wurden in Befragungen erhoben und sind in die Klassen Versorgungssicherheit und -qualität, Wirtschaftlichkeit, Einhaltung der Anforderung von Kunden in Bezug auf Umweltaspekte, Einhaltung von rechtlichen Vorgaben sowie das Schaffen eines positiven Images, eingeteilt.

Mit dem Einsatz des System Engineering Ansatzes konnten mehrere Varianten zur Optimierung des Energiekonzeptes generiert werden, Zielgrößen sind der Primärenergiebedarf und die Systemkosten. Beispielsweise existiert ein Unternehmen mit großen ungenutzten Abwärmepotential.

Ein entwickeltes Energiekonzept besteht in der Kopplung der Sektoren Wärme und Strom innerhalb des Gewerbegebietes. Dafür soll zum einen in ein gasbetriebenes Blockheizkraftwerk (BHKW) investiert werden, das Strom sowie Wärme simultan erzeugen kann. Zum anderen soll die Abwärme für die Raum- und Warmwassererwärmung verwendet werden. Dafür wird eine Wärmepumpe benötigt, um die anfallende Abwärme auf ein passendes Temperaturniveau zu erhöhen. Die Wärmepumpe soll dabei bestmöglich durch die bereitgestellte elektrische Energie des BHKWs betrieben werden.

Die Abwärme kann über ein neu zu errichtendes Nahwärmenetz an die teilnehmenden Akteure distribuiert werden. Weitere benötigte Wärme wird über die einzelnen gasbetriebenen Heizkessel generiert. Damit überschüssige Wärmeenergie zeitlich verschoben eingesetzt werden kann, soll ebenso ein Wärmespeicher installiert werden, um einen noch effizienteren Einsatz der thermischen Energie zu erreichen.

2 Validierung mithilfe von Simulation

Für die Verifizierung der Szenarien wird auf die Simulation zurückgegriffen. Damit sollen die technische Machbarkeit, die Möglichkeiten und Grenzen von Betriebsführungskonzepten und darauf aufbauend die wirtschaftliche Tragfähigkeit überprüft werden. Im Projektverlauf ergab sich zum Beispiel, dass das Unternehmen mit dem Abwärmepotential zwischenzeitlich die Produktion am Standort eingestellt hat. Ein zu entwickelndes Energiekonzept muss gegenüber solchen Änderungen robust sein und entsprechende Redundanzen vorweisen. Eine Simulation kommt laut VDI-Richtlinie 3633 Blatt 1 in Frage, da es sich um ein komplexes System handelt, das eine Vielzahl an Daten, Einflüssen und Abhängigkeiten

besitzt [3]. Die Richtlinie wird normalerweise für die Simulationswürdigkeit von Problemstellungen des Materialflusses in Logistik und Produktion herangezogen. Laut Stoldt [4] besitzt die VDI-Richtlinie 3633 Blatt 1 jedoch die Voraussetzungen für eine grundsätzliche Anwendung auf beliebige Simulationsstudien jeglichen Kontexts.

Bezogen auf das Energiekonzept sollen mithilfe der Simulationssoftware TOP-Energy realitätsnahe Energieflüsse innerhalb des Systems generiert werden, die anschließend für die Wirtschaftlichkeitsbetrachtung verwendet werden können. Das grundlegende Vorgehen der Verifizierung wird in Abbildung 2 dargestellt.

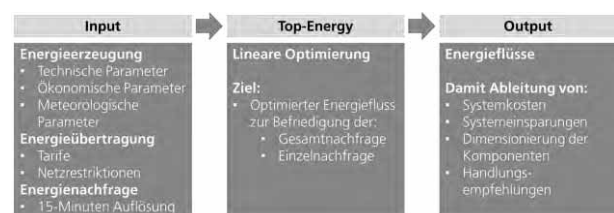


Abbildung 2: Struktur der Validierung der Energiekonzepte

Dieser Prozess befindet sich gerade in der Umsetzung und soll auch weitere Energiekonzepte validieren, um eine bestmögliche Lösung zu finden.

Angabe der Fördermittelgeber des Projekt GRIDS – Grüne Energie in industriellen Verbünden



Literatur

- [1] Industrie- und Handelskammer Chemnitz. *Gewerbe-flächenreport für den Landkreis Zwickau*. URL <https://tinyurl.com/y4nr5rb3>.
- [2] Blanchard, Benjamin S., Blyler, John E. *System Engineering Management*. Hoboken, NJ, USA: John Wiley & Sons, Inc; 2016.
- [3] VDI 3633 Blatt 1:2014-12. *Simulation von Logistik-, Materialfluss- und Produktionssystemen - Grundlagen*.
- [4] Stoldt, Johannes. *Gestaltungsmethodik für Simulationsstudien in Umplanungsprojekten zur Energieeffizienzsteigerung in Fabriken*. Chemnitz: Universitätsverlag Chemnitz; 2019.

Ein Fitnesscoach für Menschen mit Querschnittslähmung

Kilian Nickel

Fraunhofer-Institut für Intelligente Analyse- und Informationssysteme IAIS, Schloss Birlinghoven, 53757 Sankt Augustin, Deutschland; kilian.nickel@iais.fraunhofer.de

Zusammenfassung. Das Ziel des Projekts ist es, einen virtuellen Fitnesscoach für Querschnittsgelähmte zu entwickeln, der aus einer Smartphone-App und der dazu passenden sensorischen Hardware besteht. Diese Präsentation stellt das Projekt FIT-IN³ ("Ein Inklusiver, Interaktiver und Intelligenter Fitnesscoach") vor, das durch das Bundesministerium für Bildung und Forschung (BMBF) im Rahmen der Fördermaßnahme KMU-innovativ gefördert wird.

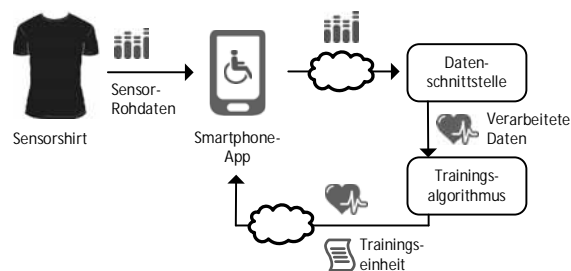


Abbildung 1: Schematische Darstellung der Komponenten des Fitnesscoach-Systems und des Datenflusses

Kontext und Ziele

Für Fitness-Begeisterte sind individuelle Trainingshilfen mittels personalisierter Smartphone-Apps, Smartwatches und anderer Wearables mittlerweile selbstverständlicher Bestandteil der normalen Trainingsroutinen. Paraplegiker*innen, also Menschen, die aufgrund einer Rückenmarksläsion querschnittsgelähmt und zur Fortbewegung auf einen Rollstuhl angewiesen sind, werden bei diesen Angeboten allerdings kaum berücksichtigt.

Jedoch ist gerade für diese Bevölkerungsgruppe kontinuierliches Training und ein Monitoring der Aktivität besonders wichtig, um Folgekrankheiten zu vermeiden. Im Bereich der personalisierbaren Smartphone-Fitness-Apps existieren kaum Angebote für Menschen mit körperlichen Einschränkungen; im Besonderen nicht sofern man den Anspruch erhebt, ohne Kompromisse in Benutzerfreundlichkeit, Praktikabilität und individueller Trainingsgestaltung im Vergleich zu Lösungen für Menschen ohne Einschränkungen trainieren zu wollen. Mit dem Projekt FIT-IN³ werden diese Lücken geschlossen und ein Demonstrator eines virtuellen Fitnesscoaches entwickelt, der, ergänzt durch Sensorik am Körper, ein umfassendes Angebot für Paraplegiker*innen darstellt. [1]

Das Vorhaben wird dazu beitragen, Paraplegikern auch ohne ständigen Zugang zu Trainern oder die Anschaffung teuren Spezial equipments ein auf die individuellen Bedürfnisse angepasstes, eigenständiges Training zu ermöglichen. Der Fokus liegt dabei auf einer Querschnittslähmung bis zum ersten Brustwirbel (Th1), also auf Rumpf und untere Extremitäten betreffende Störungen.

Das strategische Ziel des Projekts ist die Unterstützung von Menschen mit Querschnittslähmung hinsichtlich langfristiger Mobilität und Gesundheit durch sportliche Betätigung. Dazu verfolgt das Projekt folgende konkrete Ziele:

- Entwicklung einer Fitness-App (Demonstrator), die individualisierte Trainingseinheiten für Paraplegiker*innen zusammenstellt und bei der Ausführung unterstützt
- Entwicklung eines Sensorshirts zur Aufzeichnung von Vitaldaten, das in Kombination mit der App eingesetzt wird

Durch die Messung von Vitaldaten wie Pulsfrequenz und Bewegungsmustern wird eine objektive Messung physischer Aktivität ermöglicht, die zusätzlich zum Monitoring der eigenen Leistung auch in die Trainingsgestaltung der App einfließt. Die Komponenten sind schematisch in Abbildung 1 dargestellt.

1 Vorgehen und bisherige Arbeiten

Das Projekt startete im September 2019 in die Ramp-Up-Phase, in der verschiedene Grundsteine für die nachfolgenden Arbeiten gelegt wird.

Auf Grundlage von technischen, sportlichen und medizinischen Randbedingungen, sowie von wirtschaftlichen und praktischen Erwägungen wurde ein Katalog von Basisanforderungen für das Gesamtsystem aufgestellt. Ausgehend von diesen Anforderungen wurden technische Blaupausen entworfen, die die Bluetooth-Schnittstelle für die Datenübertragung zwischen Sensorshirt und Smartphone und die dazu benötigte Elektronik beschreiben. Die verbaute Sensorik wird eine Messung des Pulses ermöglichen sowie ein Erfassen bestimmter Bewegungsmuster der Sportler*innen mit Hilfe verteilter Beschleunigungssensoren. Eine der Designherausforderung besteht darin, die Elektronik so in die Textilien zu integrieren, dass die Sportausübung nicht beeinträchtigt wird.

Zur Ermittlung von Detailanforderungen der Zielgruppe wurde eine qualitative Erhebung durchgeführt, um den sportlichen Hintergrund und die körperlichen Fähigkeiten der Anwendenden, sowie ihre Erwartungen an das geplante System zu ermitteln. Gleichzeitig konnten die Verbundpartner Kontakt zu interessierten Sportlern und Sportlerinnen und Verbänden (wie z. B. dem deutschen Rollstuhl-Sportverband) aufbauen, um eine weitergehende Projektbeteiligung zu ermöglichen und letztlich eine hohe Passgenauigkeit der Ergebnisse zu erreichen.

Der Markt im Bereich Wearables – insbesondere Fitnesstracker, Smartwatches und textile Produkte (Smart Clothing) – wird laufend beobachtet, um Entwicklungen in der Sensortechnik, textiler Elektronik, datengetriebener Features und insbesondere der Schnittstellen-Strategien der Hersteller berücksichtigen zu können (Stichwort API Economy [2]). Letztere sind insofern relevant, als dass langfristig auch die Kopplung von Produkten anderer Hersteller mit der App denkbar wäre.

2 Ausblick

Die kommenden Arbeiten zielen auf die Bereitstellung eines Minimal Viable Product (MVP), bestehend aus einem Sensorshirt-Prototyp in Form eines Gurtsystems und einer ersten Version der App, ab. Mit diesem MVP wird

eine gesundheitliche Studie durchgeführt, in der zum einen die technische Funktionstüchtigkeit des Systems mit Echtdaten evaluiert wird, als auch die Tauglichkeit des Übungsrepertoires. Hier können die Teilnehmer direkt die weitere Entwicklung des Systems mitgestalten. Perspektivisch wird die finale Version des Fitnesscoaches, inklusive des konfektionierten Sensorshirts, im Q4 2022 vorliegen.

3 Die Partner

Das Vorhaben wird koordiniert durch die Kernwerk GmbH, deren bestehende Fitness-App die Grundlage für die zu entwickelnde Paraplegiker-App darstellt. Die Verbundpartner sind die ITP GmbH, die Deutsche Sporthochschule Köln, das Fraunhofer-Institut für Intelligente Analyse und Informationssysteme IAIS und die Aktion Mensch.

Fördermittelgeber

Das Projekt wird gefördert durch das Bundesministerium für Bildung und Forschung im Rahmen der Fördermaßnahme KMU-innovativ im Technologiebereich Mensch-Technik-Interaktion. [3]



Referenzen

- [1] *Webseite FIT-IN3 | Functional Fitness für Paraplegiker*innen*, <https://welcome.kernwerk.de/fit-in3>
- [2] Collins, G., Sisk, D. *API economy. From system to business services*. Deloitte Consulting LLP (Hg.), 2015. <https://www2.deloitte.com/content/dam/Deloitte/us/Documents/financial-services/us-fsi-api-economy.pdf>
- [3] *Webseite Projektinformation BMBF zu FIT-IN3*, <https://www.interaktive-technologien.de/projekte/fit-in3>

ARCH – Saving Cultural Heritage from Climate-related and Other Hazards

Daniel Lückerath*, Katharina Milde, Oliver Ullrich

Fraunhofer Institute for intelligent Analysis and Information Systems IAIS, Schloss Birlinghoven, 53757 Sankt Augustin, Germany; *daniel.lueckerath@iais.fraunhofer.de

Abstract. This presentation will introduce the EU Horizon 2020 project ARCH “Advancing resilience of historic areas against climate-related and other hazards” [1]. The presentation will introduce ARCH’s objectives, the work undertaken in the first 18 months of the project, and briefly touch on how these results go beyond the state-of-the-art.

Context and objectives

Historic towns, old urban quarters, villages, and hamlets, as well as historic landscapes make up a significant part of Europe’s identity: Natural heritage sites cover roughly 18% of the European land territory [2], and on average 22% of the European housing stock was constructed before 1946 [3]. These historic areas are deeply embedded in larger urban and rural environments, serving a role in preserving local identity and personality as well as local knowledge.

Although climate change has become one of the most significant and fastest growing threats to people and their cultural heritage [4] the impacts of climate-related and other natural hazards on historic areas have not been studied extensively [5], and disaster risk reduction seldom registers as a priority area for management of World Heritage property [6].

Therefore, there is a need for specific methods and tools that provide better information and decision-support for climate change adaptation and disaster risk reduction for heritage managers, urban planners, policy-makers, and the general public. These methods and tools need to take the unique physical, environmental, economic, social, cultural, and political aspects of historic areas, as well as the enabling conditions these areas provide for taking action into account.

However, tools and methods alone are not enough. They need to be supported by a stronger promotion of relevant public policies and participatory governance processes, which include residents from local communities and general public. The awareness of climate change

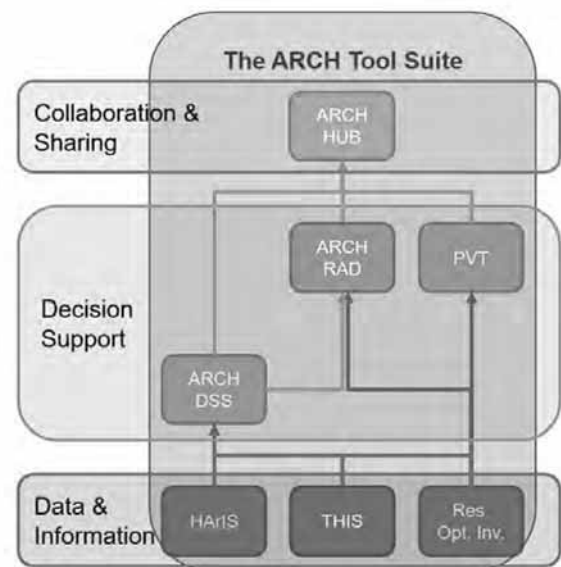


Figure 1: ARCH develops a collaborative, web-based disaster risk management platform for urban heritage districts.

impacts on historic areas needs to be increased, resilience building strategies need to be included in heritage management policies and practices, while at the same time the role heritage can play for climate change adaptation and disaster risk reduction within the wider urban context has to be emphasized.

The EU Horizon 2020 research project ARCH aims to take a step in this direction. ARCH will provide a suite of tools (see Figure 1) for assessing and improving the resilience of historic areas, and at the same time ARCH will advocate new policies as well as necessary changes to existing regulations for heritage management, climate change adaptation, and disaster risk reduction of historic areas. The ARCH solution will be targeted at heritage managers, urban planners, and other professionals from the fields of climate change adaptation and disaster risk reduction as well as politicians and the general public, in order to improve the cooperation and communication between these stakeholders and to enhance the definition and implementation of sustainable resilience building

strategies.

1 Results achieved so far

ARCH started with the production of a series of state-of-the-art reports of concepts, approaches, standards, and technologies on topics crucial for the project. These reports dealt with, amongst other things, the current state of conservation practices, the concepts of disaster risk management and building back better, gender mainstreaming, decision-support frameworks, as well as existing standards related to the ARCH's topics. Some of the key findings from these reports include:

- Cultural heritage management and disaster risk management remain poorly integrated.
- An indicator-based approach to risk assessments seems most appropriate for historic areas.
- Disaster risk management, for historic areas hinges critically on the inclusion of local communities, local practices, and traditional knowledge.
- Users need to be able to understand the decision process, limitations, and associated uncertainties of a Decision Support System (DSS).

In parallel to the state-of-the-art reports, four City Baseline Reports were compiled that establish the baseline for the municipalities of Bratislava, Camerino, Hamburg, and Valencia, with respect to their selected historic areas. As ARCH has a multi-dimensional thematic focus on 1) cultural heritage management, 2) disaster risk reduction and 3) adaptation to climate change, all three dimensions were addressed in each city's baseline review.

Based on these reports, project partners conducted match-making meetings with each city to match local problems with specific solutions provided by ARCH. These meetings, as well as all co-creation activities of the project, are governed by an ARCH co-creation guideline document that presents a harmonised methodological approach and framework for the cooperation between scientific partners and a city.

In addition to the match-making meetings, each ARCH pilot city identified 'local partnerships' and developed a work plan for local activities, aligned with the research work of the project.

ARCH also looked beyond its pilot cities and compiled a report that maps and characterises European initiatives and case studies on advancing the resilience of cultural heritage to natural and human-induced disasters as

well as to the long-term consequences of climate change. Forty initiatives are featured in this report, 32 as snapshots and eight as case studies.

In order to ensure that the solutions developed by ARCH are in line with the needs of local end-users, as expressed in the match-making sessions, and with current practices, a requirements analysis was conducted and system specifications were drawn up. ARCH partners identified 134 requirements across the four pilot cities, each mapped to one or more of the solutions that will be developed by the ARCH project partners.

Based on the exploratory work, ARCH has developed a combined disaster risk management (DRM) and climate change adaptation cycle, the ARCH Disaster Risk Management Framework. The ARCH DRM Framework is specifically aimed at historic areas and helps actors in the fields of heritage management, disaster risk management, and climate change adaptation to understand which steps are necessary to develop a combined disaster risk management and climate change adaptation plan with specific focus on historic areas.

1.1 Progress beyond the state-of-the-art

Although only halfway through the lifetime of the project, with some work packages still in the early stages of work, there is already significant impact and progress beyond the state of the art:

- The ARCH DRM Framework successfully combines the Disaster Risk Management Cycle with the Climate Change Adaptation planning cycle, also including issues of heritage management and social justice.
- The replicability framework developed for the mapping and characterisation of European initiatives and case studies provides an easily usable blueprint for identification of initiatives that might be better suited.
- The ARCH co-creation guideline provides a practical handbook for the consortium on how to conduct successful co-creation in an applied research project, which may be of use to other teams engaging in similar work.
- The establishment of local partnerships in each pilot city and the ongoing co-creation process has already resulted in higher visibility of the issues addressed by ARCH within the city administrations and larger stakeholder networks, e.g.

by establishing committed local cross-departmental partnerships that will ideally live beyond the lifetime of the project.

- The involvement of ARCH partners in formal standardisation processes on national and international level brings the project very close to influencing the state of the art in standardisation fields related to the project topics.

Acknowledgements

This paper has been prepared in the framework of the European project ARCH – Advancing Resilience of historic areas against Climate-related and other Hazards. This project has received funding from the European Union’s Horizon 2020 research and innovation programme under grant agreement no. 820999. The sole responsibility for the content of this publication lies with the authors. It does not necessarily represent the opinion of the European Union. Neither the EASME nor the European Commission are responsible for any use that may be made of the information contained therein.

References

- [1] *ARCH Advancing Resilience of Historic Areas Against Climate-related and Other Hazards*, <https://savingculturalheritage.eu/>, accessed on Jan. 26, 2021.
- [2] European Commission. *Europe’s Cultural and natural Heritage in Natura 2000*. Publications Office of the European Union. Luxembourg; 2018.
- [3] Nicol S., Roys M., Ormandy D., Ezratty V. *The cost of poor housing in the European Union*. University of Warwick; 2016.
- [4] ICOMOS, *19GA 2017/30, Resolutions of the 19th General Assembly*, Dec 2017; New Delhi, India.
- [5] Bigio A. G., Ochoa M. C., Amirtahmasebi R. *Climate-resilient, Climatefriendly World Heritage Cities*. Urban Development Series Knowledge Papers, Bd. 19, World Bank, Washington, DC, 2014.
- [6] Global Platform for Disaster Risk Reduction, *Heritage and Resilience. Issues and Opportunities for reducing disaster risks*, Global Platform for Disaster Risk Reduction. Geneva, Switzerland; 2013.

Kohärenzmodell gesunder Online-Arbeit mit exemplarischer Ableitung von Lernregeln für die Unterstützung effektiver virtueller Lernaktivitäten

Marianne Schmolke¹

¹Hochschule Wismar, Fakultät für Wirtschaftswissenschaften 23952 Wismar
marianne.schmolke@hs-wismar.de

Abstract. Die im März 2020 von der WHO aufgrund des COVID-19-Virus offiziell als Pandemie deklarierte Situation hat weitreichende Folgen gesellschaftlicher, wirtschaftlicher und für viele persönlicher Art nach sich gezogen.

Im Zuge der Pandemie war es für viele Firmen und Institutionen wie z.B. Bildungseinrichtungen notwendig, den Anteil ihrer Heimarbeit zu erhöhen, um Ansteckungen insbesondere in den Risikogruppen der Lungenkrankheit zu vermeiden. Dieses Paper fokussiert den Arbeits- und Lernalltag, der im Zuge dessen überwiegend digital stattfinden musste.

Das Angebot zur Optimierung von virtuellem Lernen und Arbeiten überfordert aufgrund nicht immer automatisch vorgegebener Struktur tendenziell eine Vielzahl derer, die versuchen, ihre Fähigkeiten systematisch zu verbessern. Das vorgestellte Konzept integriert derzeitige Erkenntnisse zum Umgang mit Effekten der zunehmenden Heimarbeit wie Zoom-Fatigue und Home-Office-Überforderung in das Kohärenzmodell nach [1] aus der gesundheits- und leistungsförderlichen Gestaltung geistiger Arbeit.

Die Erkenntnisse daraus können auszugsweise als mögliche Lernregeln dargestellt werden, die für die Konzeption von sozialen Lernplattformen verwendet werden können.

Einleitung

Die Forderung nach lebenslangem [2] und damit auch weitgehend digitalem Lernen [3] lässt sich nicht mehr wegdiskutieren. Somit bleibt es nicht nur bei der Erstausbildung oder dem Studium, sondern beim Anspruch, sich ein Leben lang nach Möglichkeiten weiterzubilden.

Die Digitalisierung bietet uns hier viele Chancen und stellt uns im Gegenzug vor eine Vielzahl von Herausforderungen, wie nachfolgend dargestellt wird.

Eine Frage, die sich stellt, ist, wie der Status Quo in der Human Resource Welt, die ja den Hauptfokus auf die Kompetenzausbildung von Arbeitskräften legt, beschrieben werden kann.

[4] stellt fest, dass wir statt den – bisherigen – Fokus auf einer ‚HR Digitalisation‘ den auf eine ‚Human Digitalisation‘ brauchen. Der Kompetenzerwerb unter

Einbezug des Individuums hat eine zentrale Bedeutung. Diese Maxime ist eine Erweiterung des Ansatzes aus der Lehre, nach dem die Lernenden mit ihrem Kompetenzerwerb und ihrer Persönlichkeit (z.B. [5]) im Mittelpunkt stehen.

Konkret umfasst der Human Digitalisation Ansatz insbesondere folgende Aspekte ([4], S. 18):

- Menschen und neue Arbeitsrealitäten in Einklang bringen sowie
- Aufbau von digitalen Kompetenzen, digitaler Kultur und entsprechender Führung in einem New-Work-Umfeld.

Kompetenzerwerb im digitalen Zeitalter ist hierzu-landes zum einen eine individuelle Angelegenheit. Dazu kann dieses Modell eine Hilfestellung liefern. Zum anderen können diejenigen, die sich beruflich mit dieser Fragestellung auseinandersetzen, das hier vorgestellte Modell nutzen, um ihr verantwortetes Lerndesign für die Lernenden zu verbessern.

Die Steuerung von Lernprozessen thematisiert [6] im Rahmen des Ansatzes von Real Time HR Analytics, d.h. der sofortigen oder zumindest besonders zeitnahen Bereitstellung und Analyse gerade entstandener Daten für Zwecke der Entscheidungsfindung. Als Beispiel führt er die Problemerkennung im E-Learning an, bei der eine Information an den Tutor übermittelt wird.

Die Verbindung eines von der ‚Human Digitalisation‘ abgeleiteten Konzepts (Kohärenzmodell für virtuelles Lernen und Arbeiten) mit den Möglichkeiten der Lerntechnologie (Learning Analytics [7], Lernflowsteuerung z.B. [8], [9]), im Speziellen: die Lernregelgenerierung, steht im Zentrum dieses Papers.

1 Grundlegende Annahmen

An dieser Stelle werden die Annahmen vorgestellt, von denen auszugehen ist, wenn eine Optimierung der Lern- und Arbeitssituation angestrebt werden soll.

Bedeutsam sind neben den generellen Erfolgsfaktoren des Online-Lernens die Aspekte:

- begünstigende Faktoren von Aufgabenwechsel sowie
- die Bewältigungsstrategie des Beharrlich-Seins.

Als generelle Erfolgsfaktoren werden die größten E-Learning-Einflussfaktoren laut der E-Learning-Management-Studie von [10] im Rahmen der COVID-19-Pandemie herangezogen:

- Blended Learning (Lernen in Präsenz – soweit möglich - und digital),
- technologisches Wissensmanagement,
- Managementunterstützung,
- Kenntnisse der Studierenden in der Nutzung von E-Learning-Systemen und
- hohe Anforderungen an die Informationstechnologie seitens der Dozierenden, Studierenden und Bildungseinrichtungen.

Diese Faktoren sollten bei grundsätzlichen Implementierungsprojekten für soziale Lernplattformen in der strategischen Ausrichtung berücksichtigt werden.

Auf der lern-individuellen Ebene sind die nachfolgenden Aspekte als Annahmen zu berücksichtigen:

1.1 Begünstigende Faktoren von Aufgabenwechsel

[11] untersuchen die Auswirkungen von Ablenkung durch z.B. soziale Medien auf den Lernerfolg. Im Ergebnis sind die Studierenden erfolgreicher, die Lernstrategien nutzen und sich in der Tendenz nicht ablenken lassen.

[12] (S. 232 f.) demonstrieren, dass Langeweile und Angst dazu führen, dass die Ressourcenaufnahmekurve abflacht, so dass eine verringerte Aufenthaltszeit an der Informationsquelle feststellbar ist, was wiederum zu schnellerem Aufgabenwechsel führe. Außerdem führe die vergrößerte Zugänglichkeit von Informationen ebenfalls zu schnellerem Aufgabenwechsel. Diesen Faktoren können die nachfolgenden Strategien entgegengesetzt

werden.

1.2 Gegenstrategien

Beharrlichkeit: [13] fand in seinen Untersuchungen Strategien, die diesen signifikanten Zusammenhängen entgegenzuwirken versprechen:

- Versuch, eine positive Stimmung beizubehalten,
- Fortschritte verfolgen,
- Sicherstellen,
 - dass das Ende der Aufgabe sowie
 - das gute Gefühl, die Aufgabe erledigt zu haben im Blick bleiben,
- sich nicht ablenken (auch nicht durch Gedanken an Dinge, jemand gerne tut).

Umgang mit Prokrastination: Ein weiterer Aspekt ist der Umgang mit Prokrastinationstendenzen z.B. [14]. Hier eignen sich Strategien, um das Belohnungssystem auszurichten (z.B. [15]).

Als Annahme für die nachfolgenden Ausführungen gelte, dass die genannten Strategien im Fokus für die Schaffung geeigneter Lern- und Arbeitsbedingungen stehen.

2 Kohärenzmodell

Das konzipierte Modell besteht in einer um die Wirkungszusammenhänge im Rahmen des virtuellen Lernens und Arbeitens erweiterten Fassung des Modells von [1] (Abbildung 1), welches in Abbildung 2 dargestellt ist.

In der vom gesundheitsförderlichen Paradigma geprägten Arbeitsgestaltung nach [1] geht es um die Gründe und Voraussetzungen, Eigenschaften und Erfahrungen, die Menschen trotz bisweilen starker Belastung gesund bleiben lassen. Der Zentralbegriff ‚Kohärenz‘ besagt, dass der Mensch ein auf Vorhersehbarkeit, Durchschaubarkeit und Verstehbarkeit basierendes Vertrauen in das, was er tun kann, was er damit erreicht sowie darauf, wie sich sein Lebensumfeld gestaltet, für seine Gesundheit benötigt. Im Zuge der zunehmenden Notwendigkeit des virtuellen Lernens und Arbeitens sind Konzepte gefordert, die dieses Modell auf den virtuellen Lernraum erweitern, um den aktuellen Herausforderungen gerecht zu werden (Abbildung 2).

2.1 Teilen von Wissen

Die erste Kohärenzebene, die im hier beschriebenen Fall

die situativen Ressourcen ‚Information‘/ ‚Kommunikation‘ mit den personalen Ressourcen ‚Selbstsicherheit‘/ ‚Selbstvertrauen‘ verbindet, kann durch das Konzept des Teilens von Wissen [16]) unterstützt werden. [16] beschreibt eine Kultur des Wissens im gesamtgesellschaftlichen Kontext. Das Teilen von Wissen beinhaltet den didaktischen Effekt, dass etwas, was erläutert oder weitergegeben wird, sehr viel intensiver durchdrungen werden muss, als wenn es nur zu verstehen ist. Die Vorhersehbarkeit, Durchschaubarkeit und Verstehbarkeit des eigenen Kommunikations- und Informationsverhaltens verstärken mit Hilfe dieses Konzepts Selbstsicherheit und Selbstvertrauen.

Optimismus und Planung werden als förderlich dargestellt – im hier erarbeiteten Modell, wird das vorausgesetzt und es wird keine konzeptuelle Ergänzung hierzu vorgenommen.

2.2 Fach- und Teamkompetenz

Fachkompetenz und Qualifikation auf der zweiten Ebene können durch Handhabbarkeit begünstigend miteinander verknüpft werden. Dazu hilft die Einbindung in ein

fachliches Netzwerk.

Arbeitsmittel und Teamkompetenz sind nicht explizit verbunden, da sie aber jeweils in den Gruppen der zweiten Kohärenzebene enthalten sind, besteht ein zumindest ungerichteter Zusammenhang.

2.3 Home Office Typen

Bezüglicher der persönlichen Arbeitstechniken und der Organisation lassen sich im Zuge der zunehmend notwendigen Heimarbeit Home Office Typen identifizieren ([17]):

- Integrierer, die Privates gut mit dem Job (oder analog der Ausbildung) kombinieren können und
- Segmentierer, die beides eher trennen.

Integrierer sollten für wichtige Besprechungen u.ä. feste Zeiten vereinbaren und Segmentierer können am besten in Gleitzeit Arbeit und Privates getrennt abhandeln ([17]).

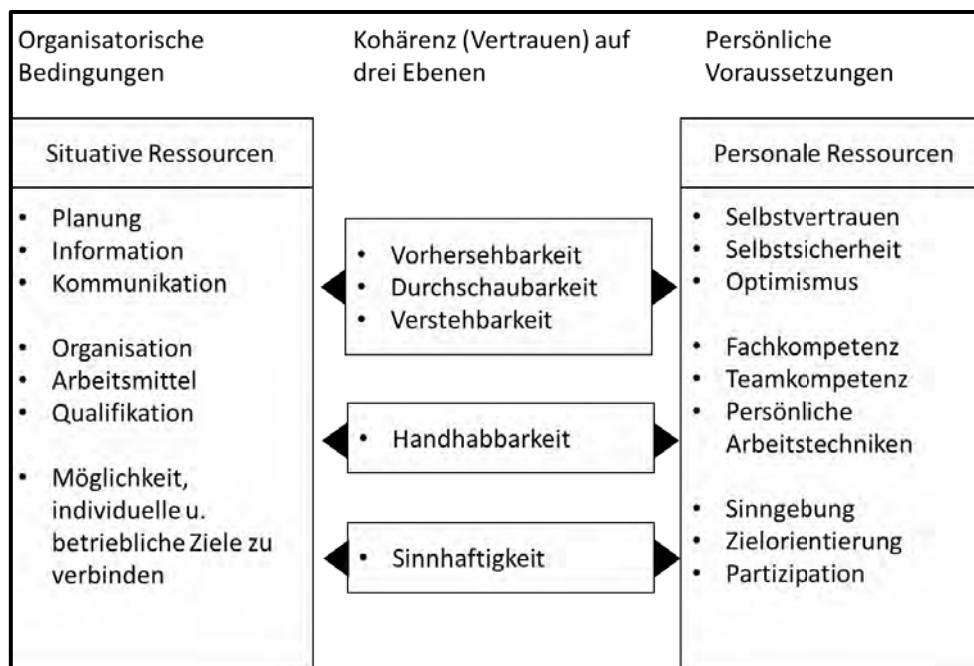


Abbildung1: Was erhält den Menschen gesund? [1] zitiert nach [15]

2.4 Zoom-Fatigue-Vermeidung

Weiterhin bedeutsam sind hier die Erkenntnisse zur Vermeidung von ‚Zoom-Fatigue‘ ([19])

- Multitasking vermeiden,
- Pausen,
- Virtuelle Social Events sollten als Opt-in möglich sein,
- Telefon und Mail bevorzugen.

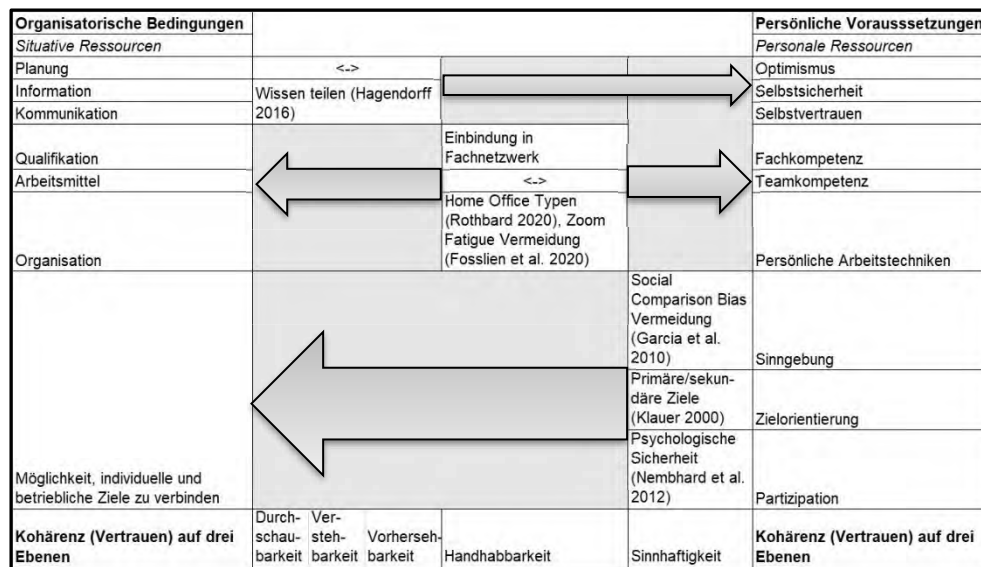
2.5 Sinn im Verbinden individueller und betrieblicher Ziele

Die Möglichkeit, individuelle und betriebliche Ziele zu verbinden, hängt daran, ob vom Individuum auf der Grundlage ihrer Sinngebung, Zielorientierung und Partizipation eine Sinnhaftigkeit wahrgenommen wird. Dies kann folgendermaßen begünstigt werden:

- Vermeidung des Social Comparison Bias ([17]): Diese Strategie zielt darauf ab, dass die eigene bearbeitete Themenstellung mehr wiegt als die Hervorhebung der eigenen Person im Vergleich zu Mitstreitern. Dies erhöht in der Konsequenz die Sinnhaftigkeit

der Tätigkeit für die Einzelperson.

- Zielerreichungsstrategien: [21] beschreibt den Planungsbeginn einer Aufgabe mit der Feststellung, welches Ziel angestrebt wird und wie dieses Ziel erreicht werden kann. [22] schlägt eine Unterscheidung in eigentliche Planungsziele (primäre Ziele, z.B. 30 Seiten für Klausur durcharbeiten) und Effizienzziele (sekundäre Ziele, z.B. nicht mehr als drei Stunden Zeit investieren) vor.
- Psychologische Sicherheit: Die Partizipation im fachlichen/ betrieblichen Netzwerk bekommt durch das Vorhandensein von psychologischer Sicherheit einen förderlichen Rahmen. Nach [23] können inklusives Verhalten von Führungskräften, gute Beziehungen in Peer Groups und die Möglichkeit, offline zu üben, psychologische Sicherheit fördern und damit eine positive Arbeitsumgebung schaffen.



Legende: Zusammenhang nicht spezifiziert <-> Ungerichtete Zuordnung im Ursprungsmodell

Zuordnung im Ursprungsmodell sowie im Zielmodell

Abbildung 2: Kohärenzmodell gesunder Online-Arbeit

3 Beispielhafte Anwendung von Lernregeln

Die hier gewonnen Erkenntnisse werden nun auf das Feld der sozialen Lernplattformen angewandt. Die Berücksichtigung der rechtlichen Bestimmungen zur Verarbeitung der personenbezogenen Daten wird bei Implementierungsvorgängen vorausgesetzt.

Es ist anzumerken, dass Regeln ausschließlich dort eingesetzt werden können, wo Vorgaben individuell realisierbar sind. Eine Kultur des offenen Wissens oder die Vermeidung des Social Comparison Bias sind Empfehlungen, die schwer regelhaft auferlegt werden können.

[9] wenden die Methode der Workflowsteuerung auf Lernprozesse an. Dieses Konzept wird nun konkretisiert. Beispielhaft lassen sich Lernregeln folgendermaßen modellieren:

Tabelle 1 und 2 zeigen Ausschnitte von Regeln, die aus logischen Überlegungen im Rahmen der Lerntheorie (z.B. [25] Erwartungs x Wert- Motivationstheorie) oder der Annahmen (1.2). Diese können in einer Lernplattform zum Beispiel, wie in [9] dargestellt umgesetzt, werden.

[24] verwenden die Metapher eines Theaterstücks, um zu veranschaulichen, wie ein Drehbuch zur Modellierung von Lerndesignmethoden verwendet werden kann. Im Hinblick auf die Lernsituation sind nicht nur objektive Kriterien wie Effektivität etc. wichtig, sondern auch, welche Kriterien der Lerndesigner als entscheidend ansieht oder nicht.

Tabelle 3 bis 4 zeigen, wie eine konkrete Lernregel-anwendung aussehen kann. Es wird das Beispiel der Home Office Typen verwendet. Effizienz und Zugänglichkeit werden für das Lerndesign-Beispiel ausgeklammert.

Process Owner Eigenschaft [Lernstrategietyp]	Process Objekt Eigenschaft [Anforderung Aufgabenstellung]	Process Kontext Eigenschaft [Verfügbarkeit Unterhaltungsmedien]	
IF			Then
Lernstrategietyp	Anforderung Aufgabenstellung	Verfügbarkeit Unterhaltungsmedien	Prozess Effektivität
schwach	-	-	gering
mittel	gering	ja	mittel
mittel	gering	nein	mittel
mittel	mittel	ja	gering
mittel	mittel	nein	mittel
mittel	hoch	ja	gering
mittel	hoch	nein	mittel
stark	gering	ja	hoch
stark	gering	nein	hoch
stark	mittel	ja	mittel
stark	mittel	nein	hoch
stark	hoch	ja	mittel
stark	hoch	nein	hoch

Tabelle 1: Schematische Lernregelmatrix 1

Process Owner Eigenschaft [Beharrlichkeitsstrategie]	Process Objekt Eigenschaft [Anforderung Aufgabenstellung]	Process Kontext Eigenschaft [Ziel-Attraktivität]	
IF			Then
Beharrlichkeitsstrategie	Anforderung Aufgabenstellung	Ziel-Attraktivität	Prozess Effektivität
schwach	-	-	gering
mittel	gering	gering	mittel
mittel	gering	mittel	hoch
mittel	gering	hoch	hoch
mittel	mittel	gering	mittel
mittel	mittel	mittel	hoch
mittel	mittel	hoch	hoch
mittel	hoch	gering	gering
mittel	hoch	mittel	mittel
mittel	hoch	hoch	mittel
stark	gering	gering	hoch
stark	gering	mittel	mittel
stark	gering	hoch	hoch
stark	mittel	gering	mittel
stark	mittel	mittel	hoch
stark	mittel	hoch	hoch
stark	hoch	gering	mittel
stark	hoch	mittel	mittel
stark	hoch	hoch	hoch

Tabelle 2: Schematische Lernregelmatrix 2

<i>If</i>	<u>Lernsituation:</u> Erforderlicher Grad an Effektivität, Effizienz, Ziel-Attraktivität, Zugänglichkeit UND Merkmale der Lernziele, Lerner, Settings, Medien UND Werte des Lerndesigners
<i>then</i>	<u>Lerndesign-Methode:</u> Eine Zusammensetzung aus einer oder mehreren aufeinanderfolgenden Handlungen mit einem oder mehreren parallelen Rollenteilen, unter Berücksichtigung einer Reihe von Bedingungen für das Szenario, die Handlung oder die Rolle
<i>with</i>	Bestimmte Erfolgswahrscheinlichkeit

Tabelle 3: Struktur von Lernregeln nach [24]

<i>If</i>	<u>Lernsituation:</u> [Erforderlicher Grad an Effektivität: Matrix 1+2] hoch, [Effizienz:] mittel, [Ziel-Attraktivität:] mittel [Zugänglichkeit:] 100% UND [Merkmale der Lernziele:] Prüfungsvorbereitung, mittelschwer, [Lerner:] Integrierer, mittlerer Lernstrategietyp, mittlere Beharrlichkeit [Settings:] Home Office, [Medien:] PC mit Internet UND [Werte des Lerndesigners:] Übungscoach
<i>then</i>	<u>Lerndesign-Methode:</u> Integrierer sollten für wichtige Besprechungen u.ä. feste Zeiten vereinbaren Auf Unterhaltungsmedien verzichten
<i>with</i>	[Matrix 1] mittel [Matrix 2] mittel

Abbildung 7: Anwendung der Struktur von Lernregeln Integrierer – Beispiel

<i>If</i>	<u>Lernsituation:</u> [Erforderlicher Grad an Effektivität: Matrix 1+2] hoch, [Effizienz:] mittel, [Ziel-Attraktivität:] mittel, [Zugänglichkeit:] 100% UND [Merkmale der Lernziele:] Prüfungsvorbereitung, mittelschwer, [Lerner:] Segmentierer, starker Lernstrategietyp, mittlere Beharrlichkeit [Settings:] Home Office, Medien: PC mit Internet UND Werte des Lerndesigners: unternehmerische(r) Mentor(in)
<i>then</i>	<u>Lerndesign-Methode:</u> Segmentierer können am besten in Gleitzeit Arbeit und Privates getrennt abhandeln. Auf Unterhaltungsmedien verzichten
<i>with</i>	[Matrix 1] hoch [Matrix 2] hoch

Tabelle 4: Anwendung der Struktur von Lernregeln Segmentierer - Beispiel

4 Fazit

Die Lern- und Arbeitswirklichkeit von Individuen abschließend in ein ausführbares Modell zu integrieren ist ein Anspruch, dem derzeit kaum jemand bis ins letzte Detail gerecht werden kann, so systematisch das Konzept auch erstellt sein mag.

Das vorliegende Paper dient daher vorrangig einer Orientierung zur Gestaltung von Bedingungen für virtuelles Lernen und Arbeiten und ersetzt nicht die individuelle Relativierung und Anpassung.

Ein Beitrag, der zu leisten angestrebt war, ist, dass die aktuellen Erkenntnisse zum virtuellen Lernen und Arbeiten in der Pandemie – unabhängig von existenziellen Themenstellungen, die Gegenstand anderer Betrachtungen sind – einen bedeutsamen Mehrwert für die zukünftige Gestaltung von sozialen Lernplattformen liefern. Es ist fraglich, ob in einer weniger kritischen Situation als der in den aktuellen Jahren 2020/21 Home Office Typen oder Zoom-Fatigue auf gleichwertiges öffentliches Interesse gestoßen wären.

Das lernende Individuum in den Mittelpunkt der Überlegungen zu technischen Optimierungen im virtuellen Lern- und Arbeitsumfeld zu stellen, ist eine Maxime aus dem ‚Human Digitalisation‘-Ansatz, der zu

Beginn dargelegt wurde. Es ist davon auszugehen, dass die verantwortungsvolle und technologisch-fortschrittliche Implementierung von operationalisierbaren Lernkonzepten zu den maßgeblichen Zukunftsaufgaben des Digitalisierungszeitalters gehören werden.

Literatur

- [1] Antonovsky, A. *Salutogenese. Zur Entmystifizierung der Gesundheit*. dgvt-Verlag, 1997.
- [2] Tippelt, R. Lebenslanges Lernen als Kompetenzentwicklung. *Lernen im Lebenslauf*. Wiesbaden: Springer VS; 2018. 105-120.
- [3] Helferich, P., Pleil, T. Lebenslanges Lernen in der Digitalisierung–Veränderung als Teil der DNA eines Unternehmens. *Arbeit 4.0 im Mittelstand*. Berlin, Heidelberg: Springer Gabler; 2019. 91-104.
- [4] Stelzer-Rothe, T. *Kompetenzen in der Hochschullehre: Rüstzeug für gutes Lehren und Lernen an Hochschulen*. Rinteln: Merkur Verlag; 2008.
- [5] Appel, W. Der Status quo des digitalen Personalmanagements in Deutschland: Ergebnisse einer Expertenbefragung; in: Appel, W., Wahler, M. Eds. *Die digitale HR-Organisation*, 2018
- [6] Strohmeier, S., Piazza, F. Eds. *Trends der Human Resource Intelligence and Analytics in Human Resource Intelligence and Analytics*. Springer, 2015. 350.
- [7] Ifenthaler, D., Mah, D. K., Yau, J. Eds. Utilising learning analytics for study success: Reflections on current empirical findings. Cham: Springer; 2019. 27-36.
- [8] Bergenthum, R., Desel, J., Harrer, A., Mauser, S. Learnflow Mining. 6. *e-Learning Fachtagung der GI DeLFI, LNI 2008*; vol. P-132: 269-280.
- [9] Holzhüter, M.; Frosch-Wilke, D.; Alonso, S. Discussion of the Benefit Potentials of Process Mining for e-Learning Processes. In: *CSEDU (2)*. 2010. S. 407-411.
- [10] Alqahtani, A. Y., Rajkhan, A. A.. E-learning critical success factors during the covid-19 pandemic: A comprehensive analysis of e-learning managerial perspectives. *Education Sciences*; 2020 10(9). 216.
- [11] Rosen, Larry D., L., Carrier, M. and Cheever; N.A. Facebook and texting made me do it: Media-induced task-switching while studying. *Computers in Human Behavior* 29.3. 2013. 948-958.
- [12] Gazzaley, A., and Rosen, L. *Das überforderte Gehirn: Mit Steinzeitwerkzeug in der Hightech-Welt*. Redline Verlag; 2017.
- [13] Hennecke, M., Czikmantori, T., Brandstätter, V. Doing despite Disliking: Self-Regulatory Strategies in Everyday Aversive Activities, *European Journal of Personality*, 2019. <https://doi.org/10.1002/per.2182>
- [14] Lohbeck, Annette, et al. Prokrastination bei Studierenden des Lehramts und der Erziehungswissenschaften. *Zeitschrift für Erziehungswissenschaft* 20.3. 2017. 521-536.
- [15] Duhigg, C. *Die Macht der Gewohnheit: warum wir tun, was wir tun*. ebook Berlin Verlag; 2012.
- [16] Hagendorff, T Open Data. In: Heesen, J., Ed. *Handbuch Informations- und Medienethik*. Stuttgart: Metzler, 2016. 227–233.
- [17] Rothbard, N. *Erfolg im Homeoffice ist eine Sache des Typs*, HBR; 2020
- [18] Spath, D., Braun, M. Grunewald, P. *Gesundheits- und leistungsförderliche Gestaltung geistiger Arbeit: Arbeitsgestaltung unter Einbeziehung menschlicher Eigenzeiten und Rhythmen*. Schmidt; 2004.
- [19] Fosslien, L, Duffy, M. W. *How to combat zoom fatigue*. Harvard Business Review; 2020.
- [20] Garcia, S. M., Song, H., Tesser, A. *Tainted recommendations: The social comparison bias*. *Organizational Behavior and Human Decision Processes*, 2010; 113(2), 97-101.
- [21] Schreblowski, S., Marcus H. Selbstkontrollstrategien: Planen, Überwachen, Bewerten. *Handbuch Lernstrategien*. Mandl, H., Friedrich, H.F., Eds, Hogrefe; 2006. 151-161.
- [22] Klauer, K. C. Planen im Alltag: *Ein wissenschaftlicher Prozess. Psychologie und Zukunft. Prognosen, Prophezeiungen, Pläne*. 2000. 171-187.
- [23] Nembhard, I, Edmondson A. Psychological safety. *The Oxford handbook of positive organizational scholarship*. 2012.
- [24] Koper, R., Tattersall, C. *Learning Design: A Handbook on Modelling and Delivering Networked Education and Training*. Berlin: Springer; 2005.
- [25] Heckhausen, J. E., Heckhausen, H. E. *Motivation and action*. Cambridge University Press: 2008.

Online-Prüfungen mit STACK Aufgaben

Gerda Fiedler¹, Birgit Gottschlich-Müller¹, Karin Melcher^{1*}

¹Fachbereich Maschinenbau und Mechatronik, FH Aachen, Goethestraße 1, 52064 Aachen, Deutschland;

*melcher@fh-aachen.de

Kurzfassung. Wir stellen hier exemplarisch STACK Aufgaben vor, die frei von der Problematik sind, welche sich durch diverse Kommunikationswege und (webbasierte) Computer Algebra Systeme (CAS) ergibt. Daher sind sie insbesondere für eine Open-Book Online Prüfung geeignet, da eine faire Prüfungssituation gewährleistet werden kann.

Einleitung

Seit dem WS 2020/21 gibt es im Curriculum aller Bachelorstudiengänge des Fachbereichs Maschinenbau und Mechatronik an der FH Aachen ein neues Modul "Grundlagen der Mathematik für Ingenieure". Dieses findet als zweiwöchiger Blockkurs vor Beginn der regulären Vorlesungszeit statt und ersetzt den zweiten Teil des früher vierwöchigen Mathematik Vorkurses. Der Großteil des Moduls widmet sich der Lösung verschiedener Gleichungstypen sowie den verschiedenen Funktionsklassen. Außerdem gibt es eine kurze Einführung in die Differential- und Integralrechnung und es werden die Ableitungsregeln besprochen. Das Bestehen dieser Klausur ist Voraussetzung zur Anmeldung der Mathematik 1 Klausur nach dem ersten Semester und aller Modulprüfungen ab dem zweiten Semester.

Eigentlich als Präsenzklausur ohne Zulassung von Hilfsmitteln (also weder Taschenrechner noch Formelsammlung) geplant, wurde sie aufgrund der Coronapandemie durch eine Online-Prüfung mit STACK Aufgaben ersetzt. Aus rechtlichen Gründen wurde diese Klausur als Open-Book Prüfung (also ohne Aufsicht) durchgeführt. Ein wichtiger Aspekt bei der Konzeption der Aufgaben war daher, eine faire Prüfungssituation zu gestalten. Aufgaben, bei denen Ableitungen zu bestimmen sind oder Gleichungen gelöst werden müssen, können sehr einfach mittels eines CAS gelöst werden, sind daher für solche Klausuren nur bedingt geeignet. Wir möchten in diesem Artikel einige unserer Konzepte vorstellen, die wir für die Online-Prüfung genutzt haben und einen Ausblick

auf geeignete Aufgaben im Bereich der Höheren Mathematik geben.

1 Was sind STACK Aufgaben?

STACK steht für "System for Teaching and Assessment using a Computer algebra Kernel" – ein webbasiertes, open-source Plugin, das direkt in Online-Tests in ILIAS bzw. Moodle eingebunden werden kann. STACK Aufgaben lassen sich sehr einfach randomisieren und durch die Einbindung von Maxima können Eingaben direkt symbolisch nachgerechnet und Folgefehler gut nachvollzogen werden. [1]

2 Der Weg ist das Ziel

Bei schriftlichen Mathematik Klausuren ist im allgemeinen der Lösungsweg von größerer Bedeutung als das reine Endergebnis. Mathematische Online Aufgaben, die nur ein Endergebnis abfragen, sind zwar meist sehr einfach zu realisieren, können aber die Kompetenz "Lösungsstrategie" nicht überprüfen. Daher bietet es sich an, Online Aufgaben zu konzipieren, wo auch Zwischenschritte angegeben werden müssen. Diese haben den weiteren großen Vorteil, dass oft auch Folgefehler berücksichtigt werden können.

Soll bei der Lösung von Gleichungen (oder Ungleichungen) nicht nur die Lösungsmenge angegeben werden, kann als Zwischenschritt beispielsweise auch nach äquivalenten Darstellungen der ursprünglichen Gleichung gefragt werden. Im Beispiel einer Logarithmusgleichung $\ln(a(x-b)^2) = c$ (vgl. Abbildung 1), wobei a , b und c randomisiert sind, gibt es verschiedene Wege diese nach x aufzulösen, je nachdem welches Logarithmusgesetz angewendet wird. Zusätzlich ist dabei stets auf den Definitionsbereich zu achten. Hier sind drei der sechs angegebenen Gleichungen äquivalent zur Ausgangsgleichung. Neben dem reinen Lösen der Logarithmusgleichung müssen die Studieren-

den hier auch zeigen, dass sie erkennen können, welche drei der sechs angegebenen Gleichungen tatsächlich äquivalent sind.

Gegeben sei die Gleichung

$$\ln(4(x-5)^2) = 2.$$

Zu welcher der folgenden Gleichungen ist sie äquivalent?

- (1) $\ln((x-5)^2) = 2 - 2 \ln(2)$ (4) $(x-5)^2 = \frac{1}{4}e^2, x \neq 5$
 (2) $\ln(x-5) = \frac{2-2 \ln(2)}{2}$ (5) $(x-5)^2 = e^{2-2 \ln(2)}, x > 5$
 (3) $\ln(x-5) = \sqrt{2-2 \ln(2)}$ (6) $|x-5| = e^{1-\ln(2)}, x \neq 5$

Geben Sie Ihre Auswahl als Menge $\{nr_1, nr_2, \dots\}$ ein.

Auswahl:

Geben Sie nun die Lösungsmenge der gegebenen Gleichung in der Form $\{x_1, x_2, \dots\}$ ein.

$\mathbb{L} =$

Abbildung 1: STACK Aufgabe: Logarithmusgleichung I.

Eine andere Möglichkeit, bei der auch Folgefehler berücksichtigt werden können, zeigt folgende Logarithmusgleichung (vgl. Abbildung 2). Ziel dieser Aufgabe ist es eine Logarithmusgleichung der Form

$$\ln(x-a) + \ln(bx-a) = 2 \ln(x+a)$$

zu lösen, wobei a und b randomisiert sind. Nach der Bestimmung des Definitionsbereiches führt die Anwendung der Exponentialfunktion auf eine quadratische Gleichung, deren Koeffizienten angegeben werden müssen. So können in der Lösungsmenge etwaige Rechenfehler bei der Umformung berücksichtigt werden.

Zu lösen ist die Gleichung $\ln(x-5) + \ln(2x-5) = 2 \ln(x+5)$.

a) Bestimmen Sie den Definitionsbereich \mathbb{D} der Gleichung.

Geben Sie zunächst die Nummer für die Form von \mathbb{D} und dann einen Wert für a ein.

- (1) $\mathbb{D} = (-\infty, a)$ (2) $\mathbb{D} = (-\infty, a]$ (3) $\mathbb{D} = (a, \infty)$ (4) $\mathbb{D} = [a, \infty)$

Form Nr. $a =$

b) Geben Sie eine quadratische Gleichung der Form $x^2 + bx + c = 0$ an, die auf \mathbb{D} äquivalent zur gegebenen Gleichung ist. Geben Sie die Parameter b und c ein.

$b =$ $c =$

c) Geben Sie schließlich die Lösungen der ursprünglichen Logarithmusgleichung als Menge $\{x_1, x_2, \dots\}$ an.

$\mathbb{L} =$

Abbildung 2: STACK Aufgabe: Logarithmusgleichung II.

Bei Grenzwertaufgaben in der Höheren Mathematik liegt ein Schwerpunkt in den Termumformungen, die es erlauben Grenzwertsätze anzuwenden. Abbildung 3 zeigt das klassische Beispiel einer Folge, die so umgeformt werden kann, dass als Grenzwert die Eulersche Zahl vorkommt. Randomisiert wird in diesem Fall nur der Faktor von n im Exponenten, der sowohl positiv als

auch negativ sein kann, aber ungleich $-1, 0$ und 1 ist.

Bestimmen Sie den Grenzwert der folgenden Folge, falls er existiert

$$a_n = \left(1 - \frac{1}{n}\right)^{6n}, \quad n \in \mathbb{N}.$$

a) Wandeln Sie zunächst das Folgenglied a_n so in die Form $\left(\left(\frac{m+1}{m}\right)^m\right)^k \cdot b_n$ um, dass die Grenzwertsätze direkt anwendbar sind.

$m =$ $k =$ $b_n =$

b) Bestimmen Sie den Grenzwert (falls er existiert)

$$\lim_{n \rightarrow \infty} b_n =$$

c) Bestimmen Sie schließlich den Grenzwert (falls er existiert)

$$\lim_{n \rightarrow \infty} a_n =$$

Hinweis: Falls ein Grenzwert nicht existiert, geben Sie bitte **inf** ein.

Abbildung 3: STACK Aufgabe: Grenzwertbestimmung.

Bei der nächsten Aufgabe (vgl. Abbildung 4) handelt es sich um ein Integral, das mittels Substitution lösbar ist. Es soll die Stammfunktion von $f(x) = \frac{bx}{\sqrt{1+ax^2}}$ ermittelt werden, wobei a und b randomisiert sind. Als Zwischenschritte müssen als erstes die gewählte Substitution angegeben werden, danach der daraus resultierende Integrand sowie dessen Stammfunktion.

Gegeben sei die Funktion $f: \mathbb{D} \rightarrow \mathbb{R}$ mit $f(x) = \frac{9x}{\sqrt{6x^2+1}}$.

Bestimmen Sie eine Stammfunktion F mit Hilfe der Substitutionsmethode.

a) Ersetzen Sie zunächst einen Teil des Integranden durch eine neue Integrationsvariable z

$z(x) =$

b) Die Substitution aus a) führt zu dem Integral $\int g(z) dz$ mit

$g(z) =$

Bestimmen Sie eine Stammfunktion von g .

$G(z) =$

c) Geben Sie schließlich eine gesuchte Stammfunktion von f an.

$F(x) =$

Abbildung 4: STACK Aufgabe: Stammfunktion mittels Substitutionsmethode ermitteln.

3 Rückwärts formuliert und mit Parametern gewürzt

Mittels "rückwärts" formulierter Aufgaben und/oder durch den Einsatz von Parametern ist die (direkte) Nutzung von Online-Tools schwierig. Abbildung 5 zeigt das Beispiel einer Bruchgleichung mit einem Parameter p

sowie einem linearen Zähler- und Nennerterm, deren Koeffizienten randomisiert werden. Bestimmt werden soll die Definitionsmenge, die Lösung x in Abhängigkeit von p sowie der Wert des Parameters p , bei der die Gleichung nicht lösbar ist.

Für $x \in \mathbb{R} \setminus M$ ist die Gleichung

$$\frac{p-2x}{5x+2} = -1$$

mit dem Parameter $p \in \mathbb{R}$ gegeben.

a) Bestimmen Sie die Menge M der Definitionslücken.

$M =$ ✓

b) Bestimmen Sie die Lösung x der Gleichung in Abhängigkeit von p .

$x =$ ✓

c) Für welchen Parameter p hat die Gleichung keine Lösung?

$p =$ ✓

Abbildung 5: STACK Aufgabe: Bruchgleichung mit Parameter.

In der folgenden Aufgabe (vgl. Abbildung 6) ist eine Wurzelfunktion $f(x) = \sqrt{p_s(x)}$ samt maximalem Definitionsbereich gegeben, wobei p_s eine konkave quadratische Funktion in Abhängigkeit eines Parameters s ist. Der Parameter s soll so bestimmt werden, dass der Definitionsbereich dem vorgegeben Definitionsbereich entspricht. Statt der Wurzelfunktion kann auch die Logarithmusfunktion verwendet werden, sofern aus dem abgeschlossenen ein offenes maximales Definitionsintervall wird.

Für welche Parameter $s \in \mathbb{R}$ hat die Funktion

$$f(x) = \sqrt{-x^2 + (2-3s)x - 2s^2 + 3s - 1}$$

den maximalen Definitionsbereich $\mathbb{D} = [-3, -1]$?

$s \in$ ✓

Abbildung 6: STACK Aufgabe: Definitionsbereich mit Parameter.

In der folgenden Aufgabe (vgl. Abbildung 7) sollen die Parameter b und c einer trigonometrischen Gleichung so bestimmt werden, dass sie auf dem Intervall $[0, 2\pi)$ genau eine Lösung besitzt. Randomisiert wird in diesem Fall nur der Vorfaktor von $\sin^2(x)$.

Für welche Parameterwerte $b, c \in \mathbb{R}$ hat die folgende Gleichung im Intervall $[0, 2\pi)$ genau eine Lösung?

$$\frac{7}{2} \sin^2(x) = b \cos(x) + c$$

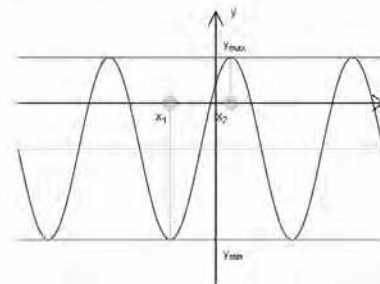
$b =$ ✓ $c =$ ✓

Abbildung 7: STACK Aufgabe: Trigonometrische Gleichung mit Parameter

4 Vom Graph zur Funktion

In diesem Aufgabentyp wird ein (randomisierter) Graph einer Funktion angezeigt und die Prüfungsteilnehmer müssen entweder die Funktionsgleichung angeben, die Parameter einer gegebenen Funktionsgleichung anpassen oder den Typ der Funktionsgleichung erkennen. In Abbildung 8 sieht man das Beispiel einer allgemeinen Cosinus Funktion $f(x) = A \cos(\omega x + \varphi) + y_0$, von der zwei Extremstellen in beliebiger Kombination aus Maxima und Minima sowie minimaler und maximaler Funktionswerte, jeweils randomisiert, bekannt sind und die Parameter A , ω , φ sowie y_0 bestimmt werden müssen.

Gegeben ist eine periodische Funktion mit der Darstellung $f(x) = A \cos(\omega x + \varphi) + y_0$:



Dabei sind gemäß der Abbildung

- $x_1 = -3\pi$ und $x_2 = \pi$ zwei ausgewählte Extremstellen sowie
- $y_{\min} = -3$ und $y_{\max} = 1$ die beiden absoluten Extremwerte von f .

Bestimmen Sie **einen Satz** von zusammen passenden Parametern A , ω , φ , y_0 .

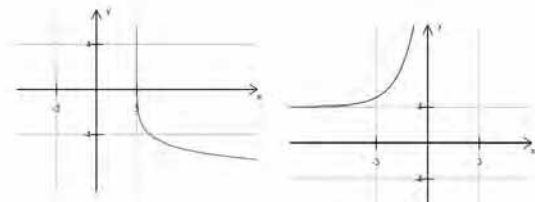
$A =$ ✓ $\omega =$ ✓
 $\varphi =$ ✓ $y_0 =$ ✓

Abbildung 8: STACK Aufgabe: Allgemeine Cosinusfunktion.

In der folgenden Aufgabe (vgl. Abbildung 9) sind jeweils die Graphen zweier linear transformierter Exponential- und/oder Logarithmusfunktionen (ohne Stauchung bzw. Streckung) sowie neun Funktionsvorschriften vorgegeben. Jedem Graphen ist die

richtige Funktionsvorschrift zuzuordnen. Die Verschiebung in x - beziehungsweise y -Richtung ist jeweils betragsmäßig gleich randomisiert, zusätzlich kann die Funktion noch an beiden Achsen gespiegelt werden.

Gegeben sind die Funktionsgraphen für $f(x)$ und $g(x)$:



sowie die Funktionsvorschriften

- | | | |
|------------------------------|-----------------------------|------------------------------|
| (1) $f_1(x) = -\ln(x-3) - 4$ | (4) $f_4(x) = e^{x-3} + 4$ | (7) $f_7(x) = -\ln(3-x) - 4$ |
| (2) $f_2(x) = e^{x+3} + 4$ | (5) $f_5(x) = e^{x+1} - 4$ | (8) $f_8(x) = 4 - \ln(x+3)$ |
| (3) $f_3(x) = e^{x+2} + 4$ | (6) $f_6(x) = 4 - \ln(x-3)$ | (9) $f_9(x) = 4 - \ln(-x-3)$ |

Geben Sie die Nummern der Funktionsvorschriften an, die den Funktionen aus den Abbildungen entsprechen:

- a) Nummer der Funktionsvorschrift für $f(x)$: ✓
- b) Nummer der Funktionsvorschrift für $g(x)$: ✓

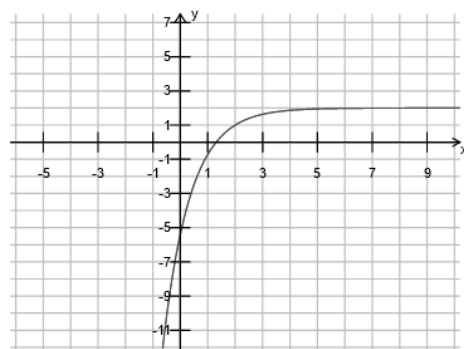
Abbildung 9: STACK Aufgabe: Exponential- und Logarithmusfunktion.

In der nächsten Aufgabe (vgl. Abbildung 10) soll als erstes erkannt werden, von welchem Funktionstyp (Wurzel-, Exponential-, oder Logarithmusfunktion) der dargestellte Graph ist. Dargestellt wird eine zufällig ausgewählte Funktion der Form $f(x) = \sqrt{x-a} + b$, $f(x) = -e^{-x+a} + b$ oder $f(x) = \ln(x-a) + b$, wobei a und b randomisiert werden. Danach sollen auch noch die Parameter a und b der ausgewählten Funktion bestimmt werden. Als Hinweis wird noch ein konkreter Punkt des Graphen angegeben.

In Abbildung 11 sieht man ein Beispiel aus dem Bereich der Höheren Mathematik zum Thema (einseitiger) Funktionsgrenzwert. Es werden jeweils zwei Stellen randomisiert, eine ist immer eine Sprungstelle (entweder als Definitionslücke oder Unstetigkeitsstelle). Von diesen Stellen sind jeweils die einseitigen Funktionsgrenzwerte zu bestimmen. Zusätzlich soll der Typ dieser Stelle angegeben werden.

5 Vom Text zur Gleichung

Da bei Textaufgaben die größte Schwierigkeit meist im Aufstellen der notwendigen Ansatz-Gleichung



Die Abbildung zeigt den Graphen einer Funktion vom Typ

- (1) $f(x) = \sqrt{x-a} + b$ oder
 (2) $f(x) = -e^{-(x-a)} + b$ oder
 (3) $f(x) = \ln(x-a) + b$.

Hinweis: Der Punkt $P(2, 1)$ liegt auf dem Graphen von f .

Geben Sie zunächst an, um welchen Funktionstypen es sich handelt, indem Sie die entsprechende Nummer eingeben:

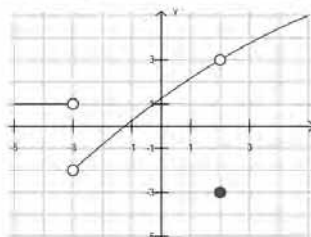
Funktionstyp Nr.: ✓

Geben Sie dann die Parameter a und b der Funktionsgleichung an:

$a =$ $b =$ ✓

Abbildung 10: STACK Aufgabe: Funktionstyp und Transformation erkennen.

Gegeben ist der Graph einer stückweise definierten Funktion $f: \mathbb{D} \rightarrow \mathbb{R}$ mit $\mathbb{D} \subset [-5, 5]$.



a) Lesen Sie die folgenden links- und rechtsseitigen Funktionsgrenzwerte ab:

$\lim_{x \rightarrow -2^-} f(x) =$ ✓ $\lim_{x \rightarrow 2^-} f(x) =$ ✓
 $\lim_{x \rightarrow -2^+} f(x) =$ ✓ $\lim_{x \rightarrow 2^+} f(x) =$ ✓

b) An welchen Stellen ...

... ist f nicht definiert? ✓ ... ist f nicht stetig? ✓
 ... hat f eine hebbare Unstetigkeit? ✓ ... ist f stetig fortsetzbar? ✓

Abbildung 11: STACK Aufgabe: einseitige Funktionsgrenzwerte.

(bzw. Gleichungen) liegt, eignen sich diese besonders gut für eine Online-Prüfung. Fordert man zusätzlich noch die Eingabe der zugrunde liegenden Gleichung, kann einerseits das „Ausprobieren“ verhindert werden. Andererseits kann ein Fehler im Endergebnis, der aus einer fehlerhaften Gleichung resultiert, als Folgefehler

erkannt werden. Durch die Randomisierung der zur Berechnung notwendigen Größen wird außerdem die Kommunikation der Studierenden untereinander erschwert. Die Herausforderung bei solchen Textaufgaben liegt in der Überprüfung der einzugebenden Gleichung, da verschiedenste Möglichkeiten berücksichtigt werden sollten. Im klassischen Füllmengenproblem (vgl. Abbildung 12) sind sowohl die benötigte Füllzeit der einen Röhre sowie das Vielfache der benötigten Zeit der anderen Röhre im Vergleich zum gemeinsamen Betrieb (jeweils randomisiert) vorgegeben.

Eine Röhre füllt einen Behälter in 6 Stunden. Eine zweite Röhre braucht dafür **fünfmal** so lang, wie wenn beide Röhren gleichzeitig geöffnet sind. Wie lange dauert es, bis beide Röhren den Behälter füllen?

Stellen Sie zu diesem Sachverhalt zunächst eine Gleichung für die Füllzeit t auf.

Gleichung:

Hinweis: Führen Sie noch **keine Umformungen** der Gleichung durch.

Lösen Sie nun die Gleichung und geben Sie die benötigte Zeit in Stunden ein.

Füllzeit:

Hinweis: Geben Sie die Zeit **ohne Angabe der Einheit** ein.

Abbildung 12: STACK Textaufgabe.

Die folgende Aufgabe (vgl. Abbildung 13) stammt aus dem Bereich der Analytischen Geometrie. Zu einem

gegebenen Vektor $\vec{x} = \begin{pmatrix} \sqrt{a} \\ 0 \\ -b \end{pmatrix}$ wird ein zweiter Vektor $\vec{y} \in \mathbb{R}^3$ mit positiver erster Komponente und vorgegebener Länge d gesucht, der senkrecht auf \vec{x} steht und mit dem zweiten Einheitsvektor einen vorgegebenen Winkel $\alpha = \frac{\pi}{3}$ einschließt. Randomisiert werden dabei a , b und d .

In der nächsten Aufgabe (vgl. Abbildung 14) ist eine Extremwertaufgabe mit mehreren reellen Variablen zu lösen. Minimiert werden soll die Oberfläche eines quaderförmigen, oben offenen Behälters bei gegebenem Volumen, welches randomisiert ist. Zuerst müssen die Formeln zur Bestimmung der Oberfläche (Hauptbedingung) und des Volumens (Nebenbedingung) angegeben werden, danach die zu minimierende Funktion und der Wert der optimalen Seitenlängen. Durch diese Formulierung können Folgefehler berücksichtigt werden.

Gegeben sei der Vektor $\vec{x} = \begin{pmatrix} \sqrt{5} \\ 0 \\ -1 \end{pmatrix}$.

Bestimmen Sie den Vektor $\vec{y} \in \mathbb{R}^3$ der folgende Bedingungen erfüllt:

(i) \vec{y} steht senkrecht auf \vec{x} .

(ii) \vec{y} hat die Länge $d = 2$ und schließt mit dem Vektor $\vec{e}_2 = \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix}$ den Winkel $\alpha = \frac{\pi}{3}$ ein.

(iii) Die erste Komponente des Vektors \vec{y} ist positiv.

Geben Sie den Vektor \vec{y} in der Form $[x, y, z]$ ein.

$\vec{y} =$ ☒

Abbildung 13: STACK Aufgabe zur Analytischen Geometrie.

Mit minimalem Materialaufwand soll ein quaderförmiger oben offener Behälter mit einem Fassungsvermögen von 5 Liter hergestellt werden. Ermitteln Sie die Seitenlängen des Behälters!

Hinweis: Länge: x cm; Breite: y cm und Höhe z cm.

(i) Geben Sie den Zusammenhang zwischen Behälteroberfläche und Seitenlängen an. (Hauptbedingung)

$O(x, y, z) =$ ☒

(ii) Geben Sie den Zusammenhang zwischen Behältervolumen und Seitenlängen an. (Nebenbedingung)

$V(x, y, z) =$ ☒

(iii) Geben Sie die Zielfunktion $O(x, y)$ in der Einheit cm^2 an.

$O(x, y) =$ ☒

(iv) Geben Sie nun die Seitenlängen des oberflächenminimierten Behälters in der Einheit cm an.

$x =$ ☒

$y =$ ☒

$z =$ ☒

Abbildung 14: STACK Extremwert Aufgabe im \mathbb{R}^2 mit Nebenbedingung.

6 Zusammenfassung

In diesem Artikel haben wir vier Typen von STACK Aufgaben vorgestellt, die für eine Open-Book Online-Prüfung im Bereich der Grundlagenmathematik konzipiert wurden und eine faire Prüfungssituation gewährleisten. Die dahinter liegenden Ansätze können, wie wir exemplarisch gezeigt haben, ohne weiteres auch für Aufgaben im Bereich der Höheren Mathematik angewendet werden. Darüber hinaus ist eine Übertragung der Konzepte auf andere Online-Prüfungssysteme möglich.

Danksagung

Eine umfangreiche Sammlung an STACK Aufgaben findet sich in "DOMAIN - Datenbank für digitale Mathematik-Aufgaben" [2].

Literatur

- [1] *Philosophy of STACK*. Link
- [2] AK MATHE-DIGITAL, *DOMAIN - Datenbank für digitale Mathematik-Aufgaben*. <http://db.ak-mathe-digital.de>

Heuristische Untersuchung von Online-Prüfungen an der TU Wien

Corinna Modiz^{1*}, Alexander Edthofer¹, Andreas Körner¹

¹Institut für Analysis und Scientific Computing, TU Wien, Wiedner Hauptstraße 8-10,
1040 Vienna, Austria; *corinna.modiz@tuwien.ac.at

Abstract. In diesem Beitrag werden die Prüfungsmodalitäten und -ergebnisse zweier Lehrveranstaltungen aus den Sommersemestern 2019 und 2020 beschrieben und verglichen. Dabei wird im Besonderen auf den Unterschied zwischen Präsenz- und Online-Prüfungen und damit verbundenen Vorurteilen eingegangen. Der Vergleich zeigt, dass eine Umstellung auf einen korrekt geplanten und didaktisch überlegten online Prüfungsmodus zu keiner signifikanten Änderung in den Ergebnissen führt.

Einführung

Seit dem Sommersemester 2020 wurde die Lehre an Universitäten vor viele neue Probleme gestellt. Die Umstellung auf Online-Lehrveranstaltungen und damit einhergehend auch Online-Prüfungen musste vollzogen werden.

Im Folgenden werden wir die Herangehensweise zu Online-Prüfen unserer Arbeitsgruppe vorstellen. Weiters werden wir die Ergebnisse der Studierenden der Prüfungen der Sommersemester 2019 und 2020 präsentieren und vergleichen, um die Auswirkungen der Distanzlehre auf die Notenverteilung zu analysieren.

Es werden die Lehrveranstaltungen „Mathematik 1 für Cluster“ und „Mathematische und statistische Grundlagen der Raumplanung“ untersucht und verglichen.

Beide Lehrveranstaltungen bestehen aus einer Vorlesung und einer begleitenden Übung, wobei eine genaue Beschreibung dieser Anteile im ersten Abschnitt folgt.

Für den Vergleich wurden ausschließlich Ergebnisse der schriftlichen Vorlesungsprüfungen im Präsenz- und Online-Modus herangezogen. Die Durchführung von Online-Prüfungen beider Lehrveranstaltungen basiert hauptsächlich auf dem Online-Kommunikationstool Zoom [1]. Auf die Prüfungsmodalitäten, Richtlinien, sowie andere verwendete Tools wird im zweiten Abschnitt näher eingegangen.

Der darauffolgende Abschnitt zeigt und vergleicht die Ergebnisse der Präsenz- und Online-Prüfungen der beider Lehrveranstaltung, die im vierten Abschnitt zusammengefasst werden.

1 Lehrveranstaltungsmodalitäten

Das Lehrkonzept, auf dem die beiden betrachteten Lehrveranstaltungen basieren, wird studienübergreifend im Bereich der Grundausbildung für Mathematik in Ingenieurwissenschaften an der TU Wien angewendet.

Sowohl die Lehrveranstaltung „Mathematische und statistische Grundlagen der Raumplanung“, als auch „Mathematik 1 für Cluster“ bestehen aus einer Vorlesung, in der der/die Vortragende den Stoff frontal vorträgt, und einer begleitenden Übung.

In der wöchentlich stattfindenden Übung der Lehrveranstaltung „Mathematische und statistische Grundlagen der Raumplanung“ werden Übungsbeispiele von dem/ der Übungsleiter_in vorgerechnet. Das dient der Wiederholung und Anwendung des Inhalts, der in der Vorlesung vermittelt wurde. Die Übung ist, im Gegensatz zur Vorlesung, nicht verbindlich zu besuchen. Die Beurteilung erfolgt auf der Basis von Klausuren, die während des Semesters abgehalten wurden.

In der begleitenden Übung des Fachs „Mathematik 1 für Cluster“ werden die vorher bereitgestellten Übungsbeispiele von Studierenden vorgerechnet. Für die Benotung werden die erreichten Punkte bei den Übungsbeispielen und den Klausuren, die zur Leistungsüberprüfung unter dem Semester dienen, herangezogen. Sowohl Vorlesung als auch Übung sind Teil der Studieneingangs- und Orientierungsphase (STEOP) und müssen daher positiv absolviert werden.

Der Beginn der COVID-19 Pandemie hatte nicht nur eine Veränderung der Prüfungsmodalitäten zur Folge. Da Präsenzlehre nicht mehr möglich war, musste der Lehrbetrieb in kurzer Zeit auf Distanzlehre umgestellt werden.

Sowohl die Lehrveranstaltung „Mathematische und statistische Grundlagen der Raumplanung“, als auch „Mathematik 1 für Cluster“ werden über das Online-Kommunikationstool Zoom abgehalten. Die Klausuren werden über das Online-System Möbius [2] (ehemalig Maple TA) abgehalten, während die Studierenden in einem Zoom-Meeting mit durchgehender Kamera- und Audioverbindung mit der Aufsichtsperson in Verbindung stehen.

Zur Klausur-, wie auch zur Prüfungsvorbereitung, wurden über Möbius verschiedene Assignments zur Vorbereitung zur Verfügung gestellt. [3]

Die Modalitäten der Klausuren sind in beiden Lehrveranstaltungen jenen der Prüfungen sehr ähnlich und werden im nächsten Abschnitt weiter ausgeführt.

2 Prüfungsmodalitäten

Während im Sommersemester des Jahres 2019 die Prüfungen der Lehrveranstaltungen „Mathematische und statistische Grundlagen der Raumplanung“ und „Mathematik 1 für Cluster“ in den Räumlichkeiten der Technischen Universität Wien abgehalten werden konnten, war diese Möglichkeit nach Beginn der Corona-Pandemie nicht beziehungsweise nur sehr eingeschränkt gegeben. Zwischen den Phasen, in denen ein Lockdown ausgesprochen wurde, waren Präsenzprüfungen zwar unter Auflagen möglich, beide Lehrveranstaltungen prüften jedoch seit Beginn der Pandemie online, um Vergleichbarkeit und Beständigkeit zu gewährleisten.

Die gemeinsame Basis des Online-Prüfens bildet Zoom, über das die Studierenden während der gesamten Prüfungszeit mit der Aufsichtsperson in Verbindung stehen. Eine dauerhafte Audio- und Videoverbindung muss gewährleistet sein. Im Vorhinein wurden Informationen zu Alternativen zu klassischen Webcams bereitgestellt. Dabei handelt es sich um Apps, die die Möglichkeit bieten, das Smart-Phone als Webcam zu benutzen. Für die funktionierende Infrastruktur während der Prüfung müssen die Studierenden eigenständig sorgen. [4]

Zur Identitätskontrolle werden die Studierenden teilweise in die Kamera gezeigt, danach die Richtlinien vorgelesen.

Studierende müssen während der gesamten Zeit allein im Raum sein und dürfen diesen nicht verlassen. Bei Verdacht, dass Studierende unerlaubte Hilfsmittel verwenden, muss der Bildschirm, unmittelbar nach der Aufforderung dazu, geteilt werden. Um Ruhe während des Arbeitens zu gewährleisten, werden Fragen über den Chat an die Aufsichtsperson gestellt.

Nach der Arbeitszeit werden die Ausarbeitung in Form eines PDFs innerhalb von 5 bis 10 Minuten in den entsprechenden TUWEL-Kurs hochgeladen. Für das Einscannen ohne passendes Gerät wurden wiederum im Vorhinein Informationen zu passenden Apps bereitgestellt.

Die Korrekturen sind dann in diesem Upload-Portal zu finden. Das gewährleistet eine erste Online-Einsichtnahme der Prüfung, wobei bei Fragen der/die Vortragende kontaktiert werden kann.

Nach erfolgreichem Hochladen der Ausarbeitung werden die Studierenden aus dem Zoom-Meeting entlassen.

Die beiden betrachteten Lehrveranstaltungen unterscheiden sich in erlaubten Hilfsmitteln und Stellung der Prüfungsfragen.

2.1 Mathematik 1 für Cluster

In der Lehrveranstaltung „Mathematik 1 für Cluster“ wird die Prüfungsangabe in Form eines PDFs zur Verfügung gestellt, das im zugehörigen TUWEL-Kurs abgelegt und erst zu einer vorher definierten Uhrzeit, also zum Prüfungsbeginn sichtbar wird. In dieser Lehrveranstaltung bestanden die Prüfungsbeispiele folglich, analog zur Präsenzprüfung, aus statischen Beispielen.

Ebenfalls analog zur Prüfung waren keinerlei Hilfsmittel zugelassen.

2.2 Mathematische und statistische Grundlagen der Raumplanung

In der Lehrveranstaltung „Mathematische und statistische Grundlagen der Raumplanung“ wird die Prüfungsangabe über das Online-System Möbius zur Verfügung gestellt. Dieses kam ebenfalls bereits in den Präsenzprüfung zum Einsatz.

Zu Beginn der Prüfung, nach Ausweiskontrolle und Verlesung der Prüfungsordnung, können die Studierenden mit einem Passwort auf die Prüfung zugreifen. Das

stellt sicher, dass erst ab Bekanntgabe des Passworts angefangen werden kann. Möbius bietet den Vorteil, dass Prüfungsbeispiele variiert werden können. Studierende bekommen folglich die gleichen Beispiele mit unterschiedlichen Zahlen, was, trotz gleichem Schwierigkeitsgrad, Möglichkeiten sich über Beispiele auszutauschen verringert.

Bei technischen Problemen hat die Prüfungsaufsicht die Möglichkeit, den/die Studierende erneut zum Prüfungsassignment in Möbius zuzulassen. Sollten Studierende also aufgrund von Verbindungsproblemen oder Problemen mit dem Browser aus dem Programm fallen, können sie ohne Zeitverlust die Prüfung fortsetzen. Die bisherigen Eingaben bleiben dabei erhalten.

Im Gegensatz zur bereits beschriebenen Lehrveranstaltung handelt es sich hier um eine Open-Book-Prüfung. Studierende dürfen während der Prüfung das Skriptum und einen nicht programmierbaren Taschenrechner als Hilfsmittel verwenden.

3 Evaluierung

Es werden nun die Ergebnisse der Prüfungen der beiden Lehrveranstaltungen aus dem Sommersemester 2020 analysiert, evaluiert und mit dem Vorjahr, als noch in Präsenz geprüft wurde, verglichen. Die Teilnehmer_innenzahlen der beiden Jahre sind mit 142 (2019S) und 126 (2020S) bei „Mathematik 1 für Cluster“ bzw. 111 (2019S) und 125 (2020S) bei „Mathematische und statistische Grundlagen der Raumplanung“ sehr ähnlich.

3.1 Mathematik 1 für Cluster

In den Abbildungen 1 bzw. 2 sind die jeweiligen prozentuellen Verteilungen der Noten der Vorlesungsprüfungen der Sommersemester 2019 bzw. 2020 abzulesen.

Während es 2019 bei insgesamt 142 abgelegten Prüfungen noch 3 Sehr gut (2,1%), 12 Gut (8,5 %), 23 Befriedigend (16,2%), 35 Genügend (24,6%) und 69 Nicht genügend (48,6%) gab, waren es bei den 126 Prüfungen des Jahres 2020 1 Sehr gut (0,8%), 13 Gut (10,3%), 19 Befriedigend (15,1%), 31 Genügend (24,6%) und 62 Nicht genügend (49,2%).

Damit sind die Ergebnisse in relativer Hinsicht ziemlich gleich. Wie bereits in Abschnitt 2.1 erwähnt, wurde die Prüfung im Jahr 2020 im gleichen Modus wie 2019

abgehalten, nur eben online. Dies brachte aber keine Veränderung der Notenverteilung mit sich.

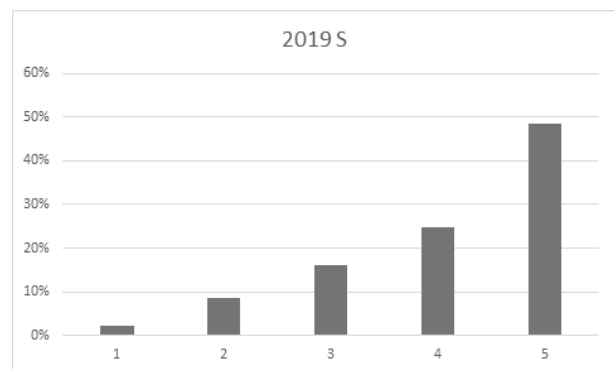


Abbildung 1: Prozentuelle Verteilung der Noten im Sommersemester 2019 von „Mathematik 1 für Cluster“.

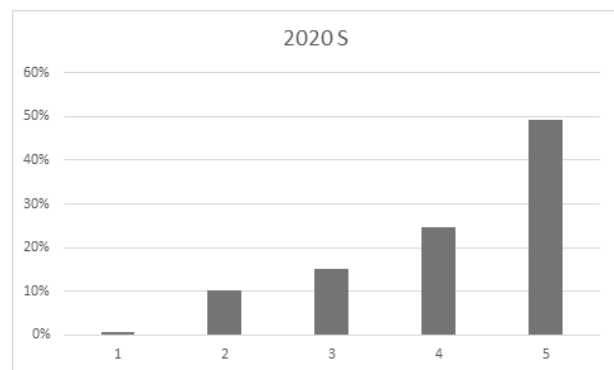


Abbildung 2: Prozentuelle Verteilung der Noten im Sommersemester 2020 von „Mathematik 1 für Cluster“.

3.2 Mathematische und statistische Grundlagen der Raumplanung

Die prozentuellen Ergebnisse der zweiten Lehrveranstaltung „Mathematische und statistische Grundlagen der Raumplanung“ sind in den Abbildungen 3 bzw. 4 für die Jahre 2019 bzw. 2020 zu finden.

Hier gab es bei den Prüfungen des Sommersemesters 2019 bei 111 Teilnehmern noch 5 Sehr gut (4,5%), 14 Gut (12,6%), 16 Befriedigend (14,4%), 30 Genügend (27%) und 46 Nicht genügend (41,4%). Im folgenden Jahr waren es bei 125 abgelegten Prüfungen 1 Sehr gut (0,8%), 9 Gut (7,2%), 18 Befriedigend (14,4%), 38 Genügend (30,4%) und 59 Nicht genügend (47,2%).

Bei Betrachtung dieser Ergebnisse fällt auf, dass sich hier eine leichte Verschlechterung der Noten zwischen

den beiden Jahren ergeben hat. In Abschnitt 2.2 wurde bereits beschrieben, dass sich bei dieser Prüfung zusätzlich zu der Umstellung auf einen Online-Betrieb der Modus dahingehend verändert hat, dass nun ein Skriptum während der Bearbeitung der Prüfung erlaubt war.

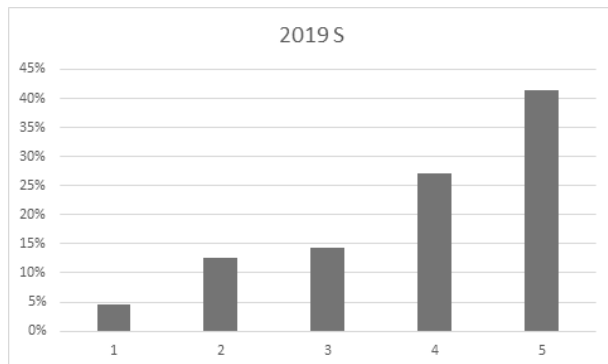


Abbildung 3: Prozentuelle Verteilung der Noten im Sommersemester 2019 von „Mathematische und statistische Grundlagen der Raumplanung“.

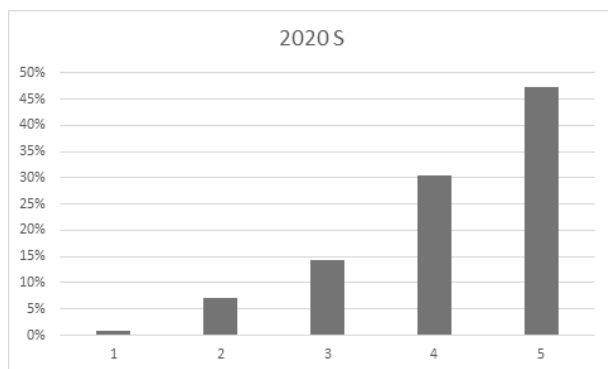


Abbildung 4: Prozentuelle Verteilung der Noten im Sommersemester 2020 von „Mathematische und statistische Grundlagen der Raumplanung“.

3.3 Vergleich der beiden Lehrveranstaltungen

Wenn die Notenverteilung der beiden Lehrveranstaltungen nun verglichen wird, so ist auffällig, dass die Ergebnisse der Prüfungen des Sommersemesters 2020 im Gegensatz zum Sommersemester 2019 gleichgeblieben bzw. sich sogar leicht verschlechtert haben.

Dies spricht gegen den „Mythos“ der Online-Prüfungen, dass Studierende viel leichter schummeln können. Bei der Lehrveranstaltung „Mathematische und statistische Grundlagen der Raumplanung“ könnte dies außerdem bedeuten, dass die Erlaubnis das Skriptum zu verwenden, sich ebenfalls negativ auswirkt, da damit das

Lernen in den Hintergrund gestellt wird und Studierende zu viel Zeit mit Blättern in den Unterlagen verbringen. Diese sogenannten „Open-Book Prüfungen“ haben also auch ihre Nachteile.

4 Conclusio und Ausblick

Die Vergleiche der Prüfungsergebnisse der beiden betrachteten Lehrveranstaltungen lassen den Schluss zu, dass die Umstellung des Prüfungsmodus aufgrund der COVID-19 Pandemie zu keiner signifikanten Verschlechterung, eigentlich zu keiner Veränderung, der Noten geführt hat.

In der Lehrveranstaltung „Mathematische und statistische Grundlagen der Raumplanung“ ist eine minimale Verschlechterung der Prüfungsergebnisse zu erkennen. Diese lässt sich jedoch auf die Umstellung auf eine Open-Book-Prüfung zurückführen und wird weiter beobachtet. Dass das Skriptum während der Prüfung verwendet werden darf, kann dazu führen, dass Studierende sich bei der Vorbereitung zu sehr auf dieses verlassen und das Suchen nach Formeln oder Erklärungen wertvolle Prüfungszeit kostet.

Eine eventuell befürchtete Verschlechterung der Ergebnisse durch die Verwendung neuer Tools wie Zoom für die Durchführung der Prüfung ist nicht eingetreten. Hierzu hat das vorherige Kennenlernen unterschiedlicher technischer Möglichkeiten in den Vorlesungen und begleitenden Übungen beigetragen.

Eine Verbesserung ist in den oben angeführten Grafiken ebenso nicht zu erkennen. Die Bedenken, der Online-Prüfungsmodus würde leichteres Schummeln mangels Kontrolle ermöglichen, haben sich in diesen Lehrveranstaltungen nicht bewahrheitet. Selbst wenn die Prüfungsangabe aller Studierenden dieselben Beispiele beinhaltet, wie in „Mathematik 1 für Cluster“, zeigt sich keine Verbesserung, die auf einen Austausch unter den Studierenden während der Prüfung hinweisen würde, in den Prüfungsergebnissen.

Um die Prüfungsaufsicht zu erleichtern, wird angedacht in den folgenden Semestern den Safe Exam Browser (SEB) zu verwenden. Diese frei verfügbare Open-Source-Software erlaubt den Benutzer_innen nur vom Prüfungspersonal vorher definierte Applikationen aufzurufen, vergleiche [5].

References

- [1] Zoom Video Communications, Inc., <https://zoom.us/>,
- [2] DigitalEd, a division of Digital Education Company Ltd.
<https://www.digitaled.com/mobius>
- [3] Körner A, Winkler S, Leskovar R, Gorgas F. Online-Komponenten der Lehre an der TU Wien. In: *Tagungsband ASIM 2018, 24. Symposium Simulationstechnik*; ISBN: 978-3-901608-12-4; 65 S.
- [4] Halbherr T, Reuter K, Schneider D, Schlienger C, Piendl T. Making examinations more valid, meaningful and motivating: the online exams service at ETH Zürich. In: *EUNIS Journal of Higher Education IT - Issue 2014/1*; ISSN: 2409-1340.
- [5] Halbherr T, Dittmann-Domenichini N, Piendl T, Schlienger C. Authentische, kompetenzorientierte Online-Prüfungen an der ETH Zürich. In: *Zeitung für Hochschulentwicklung, Jg. 11/Nr. 2*; S. 247-269.

Digitale Aufgaben und Prüfungen in Mathematik – ein Erfahrungsbericht

Karin Landenfeld^{1*}, Jonas Priebe¹, Niels Gandraß¹

¹ Fakultät Technik und Informatik, Hochschule für Angewandte Wissenschaften Hamburg, Berliner Tor 7, 20099 Hamburg; *Karin.Landenfeld@haw-hamburg.de

Einleitung

Kaum ein Studiengang lässt sich heutzutage noch vollständig ohne digitale Komponenten denken. Insbesondere die Ausrichtung der Studieninhalte auf eine zunehmend digitale Arbeitswelt ist primärer Bestandteil aktueller Strategien zur Curriculumsentwicklung an Hochschulen. Dieses bedingt, dass sowohl die Lehre und das Lernen als auch das Prüfen digital ausgerichtet werden müssen. Nur so können die neu zu erwerbenden Kompetenzen auch in geeigneter Form gelehrt und geprüft werden.

In diesem Beitrag stellen wir unsere Erfahrungen mit vorlesungsbegleitenden digitalen Übungsaufgaben zum individuellen Selbststudium vor und erläutern zwei verschiedene Szenarien zur Durchführung digitaler Prüfungen mit unterschiedlichen Randbedingungen.

1 Lernen mit digitalen Übungsaufgaben

Semesterbegleitende digitale Übungsaufgaben unterstützen sowohl die Studierenden als auch die Lehrenden.

Ein großer Vorteil für die Studierenden ergibt sich aus der Flexibilität und Individualität, die digitale Übungsaufgaben ermöglichen:

- unmittelbares Feedback direkt nach Beantwortung der Aufgaben,
- detaillierte individuelle Rückmeldung mit Hinweis auf mögliche Fehlkonzepte,
- flexible Nutzung der Aufgaben hinsichtlich Zeit, Ort und Menge,
- geeignete Wiederholbarkeit der Aufgaben.

Für die Lehrenden ergeben sich Vorteile durch eine automatisierte Korrektur sowie einer einfachen Sichtung

der Gesamt-, aber auch der Einzelergebnisse der Studierenden. Die Lehrenden können mögliche vorhandene Fehlkonzepte der Studierenden erkennen und in ihren Vorlesungen passend darauf reagieren.

Damit die digitalen Übungsaufgaben die beschriebenen Vorteile bieten, sind folgende Eigenschaften hilfreich:

- Direkte automatisierte Überprüfbarkeit einzelner Aufgaben mit sofortigem individuellem Feedback
- Berücksichtigung von Folgefehlern im Rahmen der automatischen Korrektur
- Erkennung des vorliegenden Fehlers und der Fehlkonzepte durch Implementation eines Antwortbaumes
- Individualisierung von Übungen aufgrund zufallsgesteuerter Elemente (z.B. Randomisierung von Parametern, Auswahl aus einem Fragenpool sowie Zufall in der Reihenfolge der Aufgaben)
- Wiederholte Bearbeitbarkeit der Aufgaben mit jeweils neu randomisierten Parametern
- Bereitstellung einer passenden Musterlösung zum Lernen und Verstehen

In diesem Beitrag werden exemplarisch einige Übungsaufgaben für die Mathematik 1 und 2 mit ihrer Umsetzung in Moodle¹ und STACK² vorgestellt, sowie ihr Einsatz zum vorlesungsbegleitenden Üben erläutert.

The screenshot shows a STACK question interface. At the top, it says 'Gegeben sind die beiden Matrizen A und B,' followed by the matrices $A = \begin{pmatrix} -2 & 5 & -1 \\ 3 & 4 & 2 \end{pmatrix}$ and $B = \begin{pmatrix} 2 & 5 \\ -1 & 3 \end{pmatrix}$. The question text asks: 'Welche der beiden Operationen $A \cdot B$ und $B \cdot A$ ist definiert und liefert als Ergebnis eine 2×3 -Matrix? Berechnen Sie das Produkt.' Below the text is a dropdown menu labeled 'Nicht beantwortet' and a large empty box for the answer. At the bottom left is a 'Prüfen' button.

Abbildung 1: Beispiel für eine Übungsaufgabe zur Mathematik 1: Matrix-Multiplikation mit numerischer Eingabe und Listenauswahl

¹ <https://moodle.org/>

² <http://www.stack.ed.ac.uk/>

2 Szenarien digitaler Prüfungen

Eine digitale Prüfung bildet den natürlichen Abschluss einer Veranstaltung mit vorlesungsbegleitenden digitalen Übungsaufgaben. Hierbei ist essentiell, dass die Eigenständigkeit der Durchführung unter Verwendung der erlaubten Hilfsmittel sichergestellt werden kann.

In diesem Beitrag stellen wir zwei unterschiedliche Szenarien vor, die wir beide im Wintersemester 2020/21 zur Durchführung digitaler Prüfungen in der Mathematik verwendet haben. Dies sind zum einen digitale Prüfungen in einem PC-Pool der Hochschule, in welchem die PCs mit einer eigens dafür entwickelten gesicherten Prüfungsumgebung betrieben werden. Hierbei werden alle nicht erwünschten Rechnerfunktionalitäten deaktiviert und die Internetnutzung der Studierenden wird nur geeignet gefiltert ermöglicht. Zum anderen haben wir digitale Prüfungen als Take-Home/Openbook-Prüfungen durchgeführt, in denen alle schriftlichen Materialien zugelassen waren, jedoch keine technische Absicherung und keine direkte Beaufsichtigung der Studierenden möglich war.

In beiden Prüfungsformen setzten wir eine Moodle-Lernumgebung mit der Erweiterung STACK/Maxima³ [2], [3], [4] ein und stellten eine performante Nutzung mit vielen Teilnehmern durch die an der HAW Hamburg entwickelte Maxima-Anbindung GOMaxima [1] sicher.

Beide Szenarien digitaler Prüfungen stellen jeweils vielfältige Anforderungen an die verwendeten Klausuraufgaben. Diese umfassen unter anderem:

- Die Aufgaben müssen die in der Vorlesung vermittelten Inhalte und Kompetenzen überprüfen.
- Die Inhalte der Aufgaben und die Aufgabentypen müssen sorgfältig gewählt, geplant und geprüft werden.
- Die Fragestellungen sollen eine Randomisierung ermöglichen, damit die Studierenden jeweils individuelle Aufgaben bearbeiten und kein Ergebnisaustausch möglich ist.
- Die Aufgaben sollen die Berücksichtigung von Folgefehlern und damit die Vergabe von Teilpunkten ermöglichen.
- Die Aufgaben sollen offene Fragestellungen und den Einsatz von Parametern ermöglichen.
- Insbesondere bei den Take-Home/Openbook-Prüfungen

gen ist es wichtig, reine Rechenaufgaben zu vermeiden, damit keine prüfungsexternen Rechensysteme zur Lösung verwendet werden können.



Abbildung 2: Beispiel für eine Prüfungsaufgabe zur Mathematik 1: Beurteilung der Lösbarkeit eines linearen Gleichungssystems in Abhängigkeit mehrerer Parameter mit einer Beantwortung über eine Listenauswahl

Wir stellen in diesem Beitrag beide Szenarien vor, berichten von unseren Erfahrungen und diskutieren darauf aufbauend ihre jeweilige Einsatzfähigkeit sowie ihre Grenzen.

References

- [1] Eckhoff, M., Landenfeld, K.: *GoMaxima - Eine performante Verwendung von STACK und Maxima in Lernmanagementsystemen*. In: Contributions to the 1st International STACK Conference 2018. Friedrich-Alexander-Universität: Nürnberg, Germany, 2019. DOI 0.5281/zenodo.2561190.
- [2] Kallweit, M.: *CAS-unterstütztes Assessment von Mathematik*, Computeralgebra-Rundbrief: Vol. 59. Berlin: Gesellschaft für Informatik e.V., Fachgruppe Computeralgebra, 2016.
- [3] Landenfeld, K., Göbbels, Ma., Hintze, A., Priebe, J.: *A Customized Learning Environment and Individual Learning in Mathematical Preparation Courses*. In J. Silverman & V. Hoyos (Hrsg.), Distance Learning, E-Learning and Blended Learning in Mathematics Education: International Trends in Research and Development (S. 93-111). Cham: Springer, 2018. https://doi.org/10.1007/978-3-319-90790-1_6.
- [4] Sangwin, C.: *Computer Aided Assessment of Mathematics Using STACK*. Oxford University Press, Oxford, 2013.

³ Computeralgebrasystem Maxima <http://maxima.sourceforge.net/>