

# Verlässliche und skalierbare Sicherung von Zustandsdaten verteilter cyberphysischer Systeme

Hendrik Wöhrle<sup>1</sup>, Peter Kampmann<sup>1</sup>, Frank Kirchner<sup>1,2</sup>

<sup>1</sup> Robotics Innovation Center

Deutsches Forschungszentrum fuer Kuenstliche Intelligenz GmbH  
{hendrik.woehrle, peter.kampmann, frank.kirchner}@dfki.de

<sup>2</sup> AG Robotik, Universität Bremen

**Die Datenmengen von verteilten und komplexen cyberphysischen Systemen die für die Echtzeit-Signalverarbeitung oder hochfrequente Regelung verwendet werden, z.B. in Robotersystemen, kann hinsichtlich Frequenz und Größe leicht die Datenmengen übersteigen, die von typischen Geräten im Internet der Dinge erzeugt werden. Um diese Daten für eine spätere Verwendung und Analyse verlässlich zu sichern, werden hochverfügbare Datenbanksysteme benötigt, die alle Daten persistent und flexibel speichern können. In diesem Beitrag stellen wir einen neuartigen, modellbasierten Ansatz zur Überwachung von Daten aus cyberphysischen Systemen vor, der eine hohe Verfügbarkeit und lineare Skalierbarkeit garantiert. Der Ansatz ist zunächst auf Anwendungen in der Robotik ausgerichtet, kann aber auch darüber hinaus vielfältig für Anwendungen aus dem Internet der Dinge jenseits der Robotik verwendet werden. Dies wird an einem Anwendungsbeispiel aus dem Bereich der Prädiktiven Wartung in Smart Grids erläutert.**

## 1 Einleitung

Eine Architektur, die nicht nur in Robotersystemen immer häufiger eingesetzt wird, ist der Einsatz von verteilten Regelungssystemen. In solchen Architekturen werden Sensoren und Aktoren mit (semi-)autonom arbeitenden cyberphysischen Systemen oder Rechenknoten verbunden, die eine spezifische Funktionalität implementieren können, wie z.B. lokale Sensordatenverarbeitung und -analyse oder die Steuerung bzw. Regelung von Aktoren [1]–[4]. Dadurch können diese Knoten semiautonom operieren, sind

aber in der Regel auch über ein Netzwerk mit geringer Bandbreite, z.B. ein Feldbussystem, mit einer leistungsfähigeren zentralen Recheneinheit verbunden, die zur Überwachung oder übergeordneten Steuerung eingesetzt wird. In diesen Eigenschaften ähnelt demnach die Architektur verteilter robotischer Systeme dem zunehmenden Trend, Berechnungen vermehrt aus der Cloud an den Rand des Netzwerkes, also die eingebetteten oder mobilen Systeme zu verlagern („Edge Computing“). Dieses Prinzip erlaubt es, Anwendungen die eine geringe Latenz und hohe Zuverlässigkeit voraussetzen (wie dies beispielsweise bei der Regelung von Motoren in robotischen Systemen der Fall ist), zu bedienen. Darüber hinaus weist ein solcher Ansatz Vorteile auf, wenn das Vorliegen einer Kommunikationsverbindung zu Servern bzw. Cloud-Systemen nicht garantiert werden kann oder hohe Sicherheitsanforderungen bestehen [5].

Insbesondere kann es in vielen Anwendungen wichtig sein, dass der Status eines solchen verteilten Systems aus (semi-)autonom operierenden Knoten fortlaufend überwacht wird. Dazu können die Knoten ihren Zustand mit Hilfe von sog. Telemetrienachrichten dem übergeordneten (Zentral-)System mitteilen. In bestimmten Anwendungen kann es dabei sogar der Fall sein, dass diese Telemetrienachrichten nicht explizit erzeugt werden müssen, sondern ohnehin vorliegen, z.B. wenn die Teilsysteme mit dem Zentralsystem bidirektional kommunizieren, um übergeordnete Zielvorgaben zu erhalten. Die dabei ausgetauschten Datenpakete enthalten dementsprechend dann wertvolle Informationen und sind damit hilfreich für Aufgaben wie die anschließende Analyse oder sogar auch Echtzeitüberwachung des Systemverhaltens. Eine weitere Anwendung ist die Prädiktive Wartung [6].

In diesem Beitrag stellen wir eine neuartige verteilte und hochverfügbare Datensicherungsarchitektur mit dem Namen NDLCOMDB vor, um alle Nachrichten, die in solch verteilten Steuerungssystemen erzeugt werden, zuverlässig und modellbasiert zu speichern.

Hierbei ist es von besonderer Bedeutung, dass die Datensicherung zuverlässig ist, d.h. ständig Daten entgegennehmen kann. Um die hohe Verfügbarkeit zu gewährleisten basiert NDLCOMDB daher auf der spaltenorientierten Datenbank (eng. Wide Column Store) Apache Cassandra. NoSQL-Datenbanken und insbesondere Cassandra werden häufig zur Speicherung von Daten aus dem Internet der Dinge verwendet [7], und wurden auch für Daten aus dem Betrieb von Robotersystemen genutzt [8].

Die bisherigen Arbeiten in der Robotik konzentrieren sich jedoch auf die Erfassung von anwendungsspezifischen Daten. Dadurch ist es nicht möglich,

einen solchen Ansatz für die Analyse eines Roboters aus technischer Sicht zu verwenden, d.h. um die Eigenschaften eines einzelnen Robotergelenks zu analysieren und die Veränderungen in der Leistung eines Aktors zu verfolgen. Im Gegensatz dazu zielt die hier vorgestellte Arbeit darauf ab, ein Datenbanksystem zu implementieren, das genutzt werden kann um den Zustand verschiedenartiger Systeme zu überwachen und analysieren. Daher wird in Kapitel 3.1 zusätzlich ein Anwendungsbeispiel aus dem Energiesektor diskutiert.

## 2 Architektur

Die Hauptziele unseres Ansatzes sind (1) Verfügbarkeit und Skalierbarkeit, (2) Flexibilität und (3) versionsfreies Datenhandling. Die Gesamtarchitektur gliedert sich dementsprechend in drei Ebenen (siehe Abb. 1).

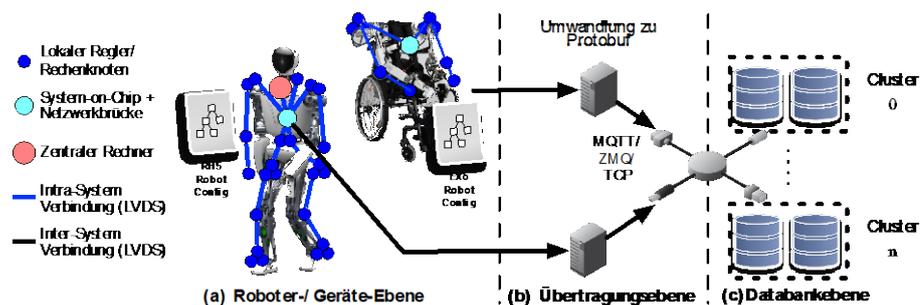


Abb. 1: Architektur-Schema von NDLCComDB.

### 2.1 Die Roboter- bzw. Geräteebene

Die physischen Hardware-Geräte, welche die Daten erzeugen, können einzelne Geräte sein, z.B. eingebettete bzw. cyberphysische Systeme, oder komplexe Objekte wie ganze Robotersysteme.

#### 2.1.1 Verteilte cyberphysische Systeme

Die in Abb. 1 gezeigten Beispielroboter bestehen beispielsweise aus mehreren Gelenken, die jeweils einen einzelnen lokalen Regler implementieren und daher zu einem gewissen Grad unabhängig voneinander bzw. (semi-)autonom arbeiten. In einem solchen verteilten System bestehen die Gelenke dazu jeweils aus einer Recheneinheit (z.B. FPGA oder Mikrocontroller), die in der Regel an mehrere Sensoren angeschlossen ist

und meistens einen einzelnen Aktor regelt. Typischerweise tauschen die Geräte jedoch ebenfalls Daten untereinander und mit einer zentralen Einheit aus, die für übergeordnete oder systembezogene Steuerungs- und regelungsaufgaben eingesetzt wird (beispielsweise Kinematik- und Dynamikberechnungen oder die komplexere modellbasierte Regelung [9].)

### **2.1.2 Kommunikation mittels NDLCOM**

Die aktuelle Implementierung von NDLCOMDB verwendet NDLCOM [10] für die Kommunikation, welches ein paketbasiertes Protokoll in den OSI-Schichten 2 und 3 implementiert und über eine serielle Vollduplex-Direktverbindung Daten austauscht. NDLCOM wurde speziell für ressourcenbegrenzte eingebettete Systeme entwickelt und erlaubt den Aufbau komplexer Netzwerktopologien, die einer Baumstruktur folgen. In NDLCOM wird jedes Gerät durch eine eindeutige ID identifiziert. NDLCOM erlaubt die Möglichkeit, Werte in bestimmte *Register* zu lesen oder zu schreiben, die sich innerhalb der Geräte befinden, die auch durch eine bestimmte *Register-ID* identifiziert werden.

Eine weitere Möglichkeit, Daten auszutauschen, sind Telemetrienachrichten, d.h. kleine Datenpakete, die von den Geräten mit einer festen Frequenz (typischerweise zwischen 10 und 1000 Hz) erzeugt werden. Telemetrienachrichten enthalten in der Regel eine Reihe von Werten, die den Zustand eines bestimmten Sensors, Aktors oder des gesamten cyberphysischen Systems enthalten. Zum Beispiel enthält eine Telemetrienachricht, die von einem Gerät erzeugt wird, welches zur Steuerung eines bürstenlosen Gleichstrommotors verwendet wird typischerweise Daten über die Position, Winkelgeschwindigkeit, Stromaufnahme etc. des zugehörigen Motors.

Das Gesamtsystem und seine Geräte werden durch eine XML-basierte Spezifikation definiert, die *RobotConfig* genannt wird (siehe Abb. 1 (a)), die systemspezifisch ist und alle Geräte des Systems auflistet. Alle Geräte gehören zu einer bestimmten Klasse, der Geräteklasse (engl. *deviceclass*), welche die Register und deren Datentyp spezifiziert. Basierend auf der *RobotConfig* können die von NDLCOM für den Datentransfer verwendeten Kommunikationsschnittstellen durch Konvertierung der XML-Spezifikation in C/C++- oder VHDL-Quellcode automatisch generiert werden.

## 2.2 Die Übermittlungsebene

Die Übermittlungsebene ist für die Übertragung der von den Geräten und Systemen erzeugten Daten in die Datenbank zuständig. Um dies zu erreichen, konvertieren wir die NDLCOM-Pakete in ein gepacktes Binärformat mit Hilfe von Google Protocol Buffers. Ähnlich wie bei den NDLCOM-Schnittstellen kann das Format mit dem RobotConfig automatisch generiert werden. Dadurch ist es möglich, verschiedene Hochsprachen für den Umgang mit den Daten und die Schnittstelle zum Datenbanksystem zu verwenden. Die Verbindung zum Datenbanksystem selbst kann über dessen (gekapselten) Datenbanktreiber erreicht werden. Zur Einbindung in größere Systeme und Unterstützung von unterschiedlichen Kommunikationsarchitekturen, z.B. dem Häufig im IoT-Bereich verwendeten Publish/Subscribe-Muster, können die Pakete optional mit verschiedenen Protokollen wie z.B. MQTT oder ZMQ, übertragen werden, welche auch die Möglichkeit bieten, eine bestimmte *Quality of Service* zu selektieren.

## 2.3 Die Datenbankebene

Um die oben genannten Anforderungen erfüllen zu können, haben wir die Apache Cassandra NoSQL Datenbank als zugrundeliegende Speichertechnologie für NDLCOMDB gewählt. Cassandra garantiert eine hohe Verfügbarkeit und Skalierbarkeit durch die zugrundeliegende verteilte Architektur, die es erlaubt, Daten zu speichern, solange mindestens einer von mehreren Knoten erreichbar ist.

Cassandra lehnt Schreiboperationen generell nicht ab, wie sie in klassischen relationalen Datenbanksystemen z.B. durch blockierte Ressourcen auftreten können, sondern ist ständig zur Annahme von Daten bereit. Die Datenbank kann auf mehrere Knoten verteilt werden, die dem Cluster dynamisch hinzugefügt oder entfernt werden können und somit eine lineare Skalierbarkeit bieten und konfigurierbare Redundanz. Dies bedeutet, dass die Daten mehrfach repliziert auf unterschiedlichen Rechnern des Clusters abgelegt werden. Durch die Gliederung in unterschiedliche Datenzentren kann zusätzlich eine räumlich getrennte Redundanz erlangt werden.

Ein weiterer Vorteil ist die Möglichkeit, Daten ohne vordefiniertes Schema zu speichern, d.h. Beziehungen zwischen verschiedenen Entitäten müssen in einer Vorentwurfsphase nicht oder nur in geringem Maße explizit modelliert werden, die Flexibilität bietet, indem sie es erlaubt, alle Tabellen nach dem

RobotConfig zu generieren. Diese Angabe kann neben den Daten gespeichert werden, was es ermöglicht, die Daten abzurufen, ohne die Struktur und Version des Systems zu kennen, zu dem die Daten gehören, da alle Datenstrukturen anhand der in der Datenbank gespeicherten Informationen erzeugt werden können.

### 3 Anwendungsbeispiele

Im Folgenden werden zwei Anwendungsbeispiele aus dem Bereich der Systemoptimierung und prädiktiven Wartung dargestellt, in denen NDLCOMDB eingesetzt wird bzw. eingesetzt werden soll.

#### 3.1 Statusüberwachung eines Exoskelettes für die Rehabilitation von Schlaganfallpatienten

Die Bedeutung und das Potential von roboterassistierter Rehabilitation in Form von Exoskeletten hat insbesondere vor dem Hintergrund des demografischen Wandels und den sich verbessernden technischen Möglichkeiten in den letzten Jahren stark zugenommen [11], [12]. Besondere Bedeutung kommt hierbei die Integration von Biosignalen, beispielsweise in Form eines Brain Computer Interfaces, in die Gesamtregelung des Systemes zu [13]. Für eine probandenspezifische Kalibrierung des Systemes ist jedoch eine Datenbank notwendig, die einen strukturierten und effizienten Zugriff auf Benutzerdaten und Modelle des Exoskelettes erlaubt. Sowohl bei den Betriebsdaten des Exoskelettes, die in Form der in Abschnitt 2.1 beschriebenen Telemetrienachrichten vorliegen, als auch bei den physiologischen Daten der Patienten handelt es sich um Zeitreihendaten. Zur späteren Analyse und Modellbildung müssen die Biosignaldaten des Benutzers *synchronisiert* mit den Betriebsdaten des Exoskelettes in der Datenbank gesichert werden, um anschließend mit Hilfe von maschinellen Lernverfahren geeignete Gesamtmodelle des Systemes mit Benutzerinteraktion erstellen zu können [14]. Für diese Aufgaben bietet NDLCOMDB ein sicheres und zuverlässiges Datenbanksystem, welches direkt mit Softwareframeworks für maschinelle Lernverfahren für die Analyse und Modellbildung gekoppelt werden kann [15].

## 3.2 Zustandsüberwachung und prädiktive Wartung in Energienetzen

Eine Anwendung außerhalb des Bereichs der Robotik ist die prädiktive Wartung und Lastverteilung in *Smart Grids*. Ein aktuelles Ziel bei der Entwicklung zukünftiger Stromnetze ist es, die Widerstandsfähigkeit und Stabilität des Energienetzes zu erhöhen und die Einbeziehung eines steigenden Anteils erneuerbarer Energiequellen zu unterstützen [16]. Das zukünftige Energienetz wird dabei eine zellulare Struktur mit verschiedenen sog. *Assets*, d.h. Anlagen, in den Zellen haben. Hierbei kann es sich um Anlagen handeln, welche die Infrastruktur des Netzes bereitstellen, oder aber um Erzeuger bzw. Verbraucher. Um die Leistungsfähigkeit des Netzes zu erhöhen soll das gesamte Netz bzw. die vorhandenen Anlagen durch intelligente Informations- und Kommunikationstechnologie überwacht und gesteuert werden, um z.B. die lokale und relativ unvorhersehbare Stromproduktion durch erneuerbare Energien und den Verbrauch durch verschiedene Konsumenten auszugleichen, d.h. bidirektionale Energieflüsse zwischen und innerhalb von Netzzellen zu steuern. Eine weitere wichtige Technologie ist die prädiktive Wartung, die eingesetzt werden kann um Systemausfälle zu minimieren und die Wartungsoperationen unter wirtschaftlichen und sicherheitstechnischen Aspekten zu optimieren.

Um diese Techniken einsetzen zu können, wird eine geeignete Informations- und Kommunikationstechnologie (IKT) benötigt, wie sie in Abb. 2. dargestellt ist. Diese muss es erlauben, Methoden der künstlichen Intelligenz lokal einsetzen zu können, um insbesondere Anforderungen hinsichtlich Latenz und Sicherheit zu genügen. Die im Vorhaben FUSE [17] geplante IKT-Infrastruktur für verteilte Datenverarbeitung wird nach den Prinzipien des Edge- und Fog-Computings entwickelt und zielt auf die Überwachung des Mittel- und Niederspannungsnetzes ab. Auf der Ebene 1 kann der jeweils durch eine Anlage fließende Strom mit geeigneten Sensoren gemessen werden.

Üblicherweise werden die gemessenen Daten nach dem Standard IEC 61850 an übergeordnete IT-Systeme übermittelt, entweder in Form von Rohdaten (IEC 61850-9-2) oder in Form von sog. GOOSE-Events (IEC 61850-8-1). GOOSE-Events können einfache Informationen über einfache Zustandsänderungen in den Assets enthalten. Für zukünftige Anwendungen und komplexe Analysen ist es jedoch notwendig, die Anlagen lokal mit Hilfe von maschinellen Lernverfahren zu überwachen und dabei beliebig komplexe

Analysen mit Hilfe von maschinellen Lernverfahren durchzuführen. Besondere Vorkommnisse werden dann an die Ebene 2 übermittelt, welche die Ereignisse vieler Ebene-1-Geräte fusioniert und eine übergeordnete Bewertung durchführt, wobei ebenfalls Methoden des maschinellen Lernens eingesetzt werden und weiter an Ebene 3 zur Benutzerinteraktion übermittelt werden.

Insbesondere in der Einführungsphase, d.h. während die Modelle, welche später in Ebene 1 genutzt werden sollen, mit Hilfe von maschinellen Lernverfahren trainiert werden, ist es notwendig, dass zunächst alle Rohdaten nach IEC 61850-9-2 verlässlich gesichert werden.

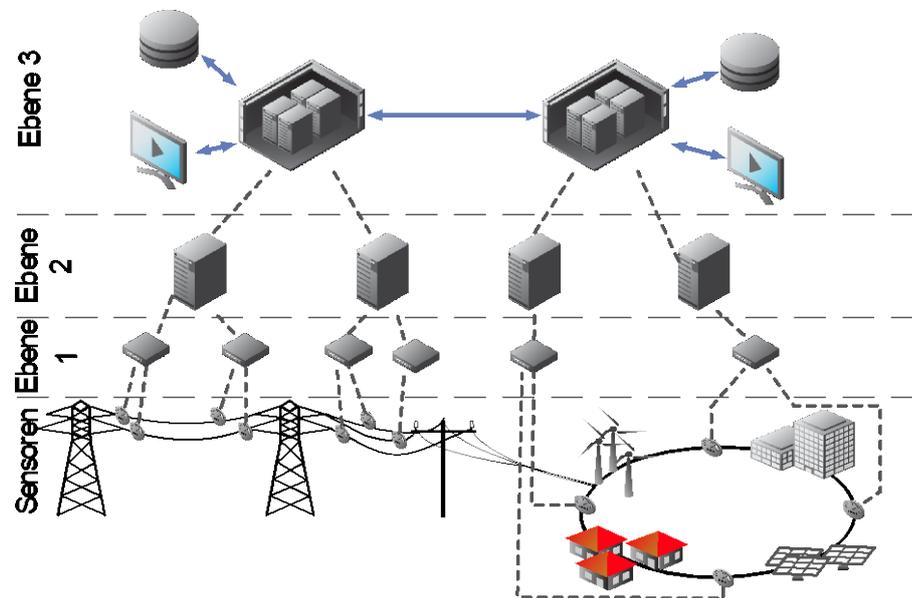


Abb. 2: Geplante Informations- und Kommunikationsarchitektur von FUSE.

Diese Rohdaten stellen dann die Trainingsdaten dar, welche für den Einsatz maschineller Lernverfahren unabdingbar sind. IEC 61850 stellt hierbei ein feste Formate zur Verfügung, welche direkt in Telemetrienachrichten für NDLCOMDB übersetzt und daher problemlos verwendet werden können.

## 4 Schlussfolgerungen und Zusammenfassung

NDLCOMDB stellt einen hochverfügbaren, linear skalierbaren Ansatz zur Überwachung verschiedener cyberphysischer oder ganzer Robotersysteme

zur Verfügung. Aufgrund seiner Flexibilität erlaubt NDLCOMDB es, alle Daten, die zwischen verteilten Rechenknoten und einem zentralen System ausgetauscht werden, in einer Datenbank zu speichern. Alle Kommunikationsschnittstellen und die Datenbanktabellen können über eine XML-basierte Spezifikation automatisch generiert werden.

In Zukunft werden wir unseren Ansatz durch die Einführung von *eingebetteten Datenbanken* erweitern. Das heißt, wir wollen den Knoten und autonom operierenden Teilsystemen um lokale Datenspeicherfähigkeiten erweitern, um alle Sensor- oder Statusdaten, die lokal erzeugt werden, zwischenspeichern zu können. Die lokal gespeicherten Daten können dann z.B. mit der NDLCOMDB synchronisiert werden, wenn sich das System später im Ruhezustand befindet. Dies würde es erlauben, nicht nur Telemetriemessages zu sichern, die nur einen Ausschnitt der Wirklichkeit bilden, sondern eine vollständige Sicherung aller Betriebsdaten zu erhalten.

## 5 Referenzen

- [1] M. Jäntschi, S. Wittmeier, und A. Knoll, „Distributed control for an anthropomorphic robot“, in *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2010, S. 5466–5471.
- [2] D. Kühn u. a., „Distributed Computation in a Quadrupedal Robotic System“, *Int. J. Adv. Robot. Syst.*, Bd. 11, Nr. 7, S. 110, 2014.
- [3] N. A. Radford u. a., „Valkyrie: NASA’s First Bipedal Humanoid Robot“, *J. Field Robot.*, Bd. 32, Nr. 3, S. 397–419, 2015.
- [4] S. Bartsch u. a., „Development and Control of the Multi-Legged Robot MANTIS“, in *Proceedings of ISR 2016: 47th International Symposium on Robotics*, 2016, S. 1–8.
- [5] N. Fallenbeck und C. Eckert, „IT-Sicherheit und Cloud Computing“, in *Industrie 4.0 in Produktion, Automatisierung und Logistik*, Springer, 2014, S. 397–431.
- [6] R. Sipos, D. Fradkin, F. Moerchen, und Z. Wang, „Log-based predictive maintenance“, in *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2014, S. 1867–1876.
- [7] S. Amghar, S. Cherdal, und S. Mouline, „Which NoSQL database for

IoT Applications?“, in *2018 International Conference on Selected Topics in Mobile and Wireless Networking (MoWNeT)*, 2018, S. 131–137.

[8] A. Dietrich, S. Zug, S. Mohammad, und J. Kaiser, „Distributed management and representation of data and context in robotic applications“, in *Intelligent Robots and Systems (IROS 2014), 2014 IEEE/RSJ International Conference on*, 2014, S. 1133–1140.

[9] D. Nguyen-Tuong und J. Peters, „Model learning for robot control: a survey“, *Cogn. Process.*, Bd. 12, Nr. 4, S. 319–340, 2011.

[10] M. Zenzes, P. Kampmann, T. Stark, und M. Schilling, „NDLCom: Simple protocol for heterogeneous embedded communication networks“, in *Proceedings of the Embedded World Exhibition & Conference, Nuremberg, Germany*, 2016, S. 23–25.

[11] H. S. Lo und S. Q. Xie, „Exoskeleton robots for upper-limb rehabilitation: State of the art and future prospects“, *Med. Eng. Phys.*, Bd. 34, Nr. 3, S. 261–268, 2012.

[12] R. Gopura, D. Bandara, K. Kiguchi, und G. K. Mann, „Developments in hardware systems of active upper-limb exoskeleton robots: A review“, *Robot. Auton. Syst.*, Bd. 75, S. 203–220, 2016.

[13] A. A. Frolov u. a., „Post-stroke rehabilitation training with a motor-imagery-based brain-computer interface (BCI)-controlled hand exoskeleton: a randomized controlled multicenter trial“, *Front. Neurosci.*, Bd. 11, S. 400, 2017.

[14] E. A. Kirchner u. a., „Recupera-Reha: Exoskeleton Technology with Integrated Biosignal Analysis for Sensorimotor Rehabilitation“, in *2. Transdisziplinäre Konferenz „Smart Assist“, December 12-13, Hamburg, Germany*, 2016, S. 504–517.

[15] M. M. Krell u. a., „pySPACE—a signal processing and classification environment in Python“, *Front. Neuroinformatics*, Bd. 7, S. 40, 2013.

[16] X. Fang, S. Misra, G. Xue, und D. Yang, „Smart grid—The new and improved power grid: A survey“, *IEEE Commun. Surv. Tutor.*, Bd. 14, Nr. 4, S. 944–980, 2012.

[17] H. Wöhrle und T. Vögele, „FUSE - FUture Smart Energy“. [Online]. Verfügbar unter: <https://robotik.dfki-bremen.de/de/forschung/projekte/fuse.html>.

Hendrik Wöhrle arbeitet als Projektleiter und stellvertretender Teamleiter am Robotics Innovation Center des Deutschen Forschungszentrums für Künstliche Intelligenz (DFKI) in Bremen. Nach den Studien der Bioinformatik und Elektrotechnik promovierte er an der Universität Bremen in der Informatik mit einer Arbeit über Datenflussarchitekturen für maschinelles Lernen mit rekonfigurierbarer Logik. Seine Forschungsschwerpunkte liegen in den Bereichen Eingebettete Systeme, Hardwarearchitekturen, Edge-Computing, maschinelles Lernen, Robotik und Biosignalverarbeitung.