

Rekonfigurierbare Datenflussarchitekturen in der Robotik

Zukünftige robotische Systeme benötigen dezentrale und verteilte Rechenarchitekturen für Intelligenz und Autonomie

Hendrik Wöhrle und Frank Kirchner, Universität Bremen und Robotics Innovation Center des Deutschen Forschungszentrums für Künstliche Intelligenz

Autoren

Hendrik Wöhrle ist wissenschaftlicher Mitarbeiter am Robotics Innovation Center und Doktorand an der Universität Bremen. Sein Forschungsschwerpunkt sind rekonfigurierbare Rechnerarchitekturen für die Echtzeitdatenverarbeitung und maschinelles Lernen.

Prof. Dr. Frank Kirchner ist Leiter der Forschungsgruppe Robotik der Universität Bremen, Direktor des Robotics Innovation Center und des Brazilian Institute of Robotics sowie Standortsprecher des Deutschen Forschungszentrums für Künstliche Intelligenz in Bremen.

Kontakt

hendrik.woehrle@dfki.de
www.dfki.de/robotik

Intelligente und autonome Roboter sind für die Entwicklung der Industrie 4.0 von essentieller Bedeutung. Sie werden als direkte Interaktionspartner gemeinsam mit dem Mensch Aufgaben lösen und Arbeiten verrichten, die wesentlich komplexer als die heutigen Aufgaben typischer Industrieroboter sind. Dabei müssen sie sich in einer unübersichtlichen und unvorhersehbaren Umwelt zurechtfinden und auf Ereignisse sofort reagieren können. Um diese Umwelt erfassen zu können und Handlungen zu planen, ist die Echtzeitverarbeitung komplexer Sensorinformationen notwendig. Herkömmliche Rechnerarchitekturen stellen sich hierbei als unzureichend heraus. Daher werden am DFKI RIC Hardware-Beschleuniger für die Robotik auf Basis des Datenflussparadigmas entwickelt.

Robotik in der Industrie 4.0

Die Industrie 4.0 beruht wesentlich auf aktuellen Entwicklungen auf den Gebieten der Cyber-Physischen Systeme, dem Internet der Dinge, dem Internet der Dienste und der Smart Factory, die zusammengefasst und miteinander kombiniert ein größeres Ganzes ergeben [1]. Hier weisen intelligente Robotik und Industrie 4.0 eine bemerkenswerte Parallelität auf, sodass aktuell mit dem wachsenden Interesse an Industrie 4.0 vielerorts Technologien und Methoden in den Fokus der Aufmerksamkeit rücken, die in der Robotik schon seit längerem aktiv erforscht werden. Beispiele sind hier Entwicklungen in den Bereichen Cyber-Physische Systeme sowie Maschine-Maschine-Interaktion, welche mit wenig Aufwand von der Robotik in Industrie 4.0-Anwendungen übertragen werden können. An einer weiteren Stelle wird aktuell gerade aktiv geforscht: die Interaktion von Mensch und Roboter in hybriden sozialen Teams [2]. Hier arbeiten Mensch und Roboter Hand in Hand zusammen, um gemeinsam komplexe Aufgaben zu erfüllen – eine essentielle Anforderung für viele Szenarien der Industrie 4.0. Die Roboter agieren hierbei *langzeitautonom* in einer komplexen Umwelt als direkter Interaktionspartner des Menschen. Roboter können hierbei in unterschiedlichsten

Formen auftreten: etwa als robotischer Manipulator, als humanoider Roboter, der ein direktes Gegenüber darstellt, oder als Exoskelett, d. h. als tragbarer Roboter, der dem Menschen beispielsweise Kraftunterstützung bietet.

Dies führt zu völlig neuen Anforderungen an die Robotik, die wesentlich schwieriger zu erfüllen sind als die Anforderungen der klassischen Produktionsrobotik. Damit Roboter mit Menschen in ihrer Umgebung interagieren können, müssen diese verlässlich und sicher auf ihre Umwelt und den Interaktionspartner Mensch reagieren. Um dies zu ermöglichen, müssen Roboter ihre Umwelt in allen Details wahrnehmen können, ein internes Bild ihrer Umgebung generieren und auf Basis dieses internen Abbilds über angemessene Handlungen entscheiden. Alle diese Schritte benötigen komplexe und rechenintensive Algorithmen, die in Echtzeit ausgeführt werden müssen:

- Eine umfassende Wahrnehmung der Umgebung benötigt die Kombination bzw. Fusion verschiedener Sensormodalitäten, beispielsweise mittels Laserscanner, Stereokameras und Nahfeldradar. Diese Sensoren erzeugen einen permanenten Strom extrem großer Datenmengen, der fortwährend verarbeitet werden muss, um die Rohdaten in verwertbare Informationen zu verwandeln. Schon herkömmlich verwendete Verfahren weisen einen hohen Rechenaufwand auf. Es zeichnet sich ab, das zukünftige Verfahren, z. B. aus dem Bereich des Deep Learnings, einen weiteren Anstieg des Rechenaufwands bedeuten.
- Damit Roboter tatsächlich die Forderung nach Intelligenz erfüllen, ist es notwendig, dass diese langzeitautonom agieren können. Sie müssen sich stetig an ihre Umgebung anpassen und *dazulernen*. Dies erfordert wiederum den Einsatz komplexer Lernalgorithmen, die *parallel zu den sonstigen aktuellen Aufgaben* ausgeführt werden.

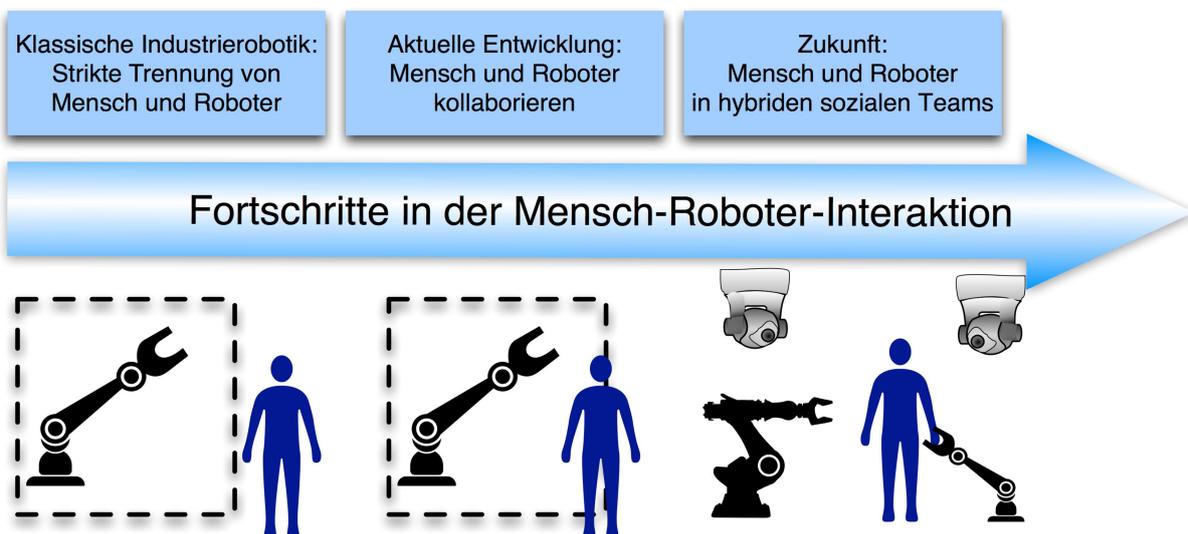


Bild 1: Roboter werden in Zukunft als direkte Interaktionspartner des Menschen agieren, d. h. nicht wie in der Vergangenheit in streng getrennten Bereichen.

Besondere Anforderungen in der Robotik

Hier wird ersichtlich, dass Eigenschaften vorliegen, die aus dem Bereich *Big Data* als die „Vier V“ bekannt sind: Volume (es werden extrem große Datenmengen erzeugt), Velocity

(die Datenmengen werden mit großer Geschwindigkeit erzeugt, Ergebnisse sollen so schnell wie möglich vorliegen), Variety (die Daten können unterschiedlichster Natur sein) und Veracity (die Daten können von schlechter Qualität sein) [3]. Hierbei weist das Gebiet der Robotik eine Reihe von Besonderheiten auf, die den Einsatz von Cloud-basierten Lösungen, die sonst in der Industrie 4.0 üblich sind, einschränken oder teilweise sogar unmöglich machen – es handelt sich um *Mobile Big Data*: (1) Roboter müssen stets *harten Echtzeitanforderungen* genügen, jegliche Verzögerung in der Datenverarbeitung stellt ein potenzielles Risiko für den Menschen dar. Daher kann die Datenverarbeitung nicht in leistungsfähige Rechner in einem Backend ausgelagert werden, um das Risiko von Verzögerungen in der Kommunikation und Verbindungsabbrüche auszuschließen. (2) Die Roboter müssen auch, zumindest streckenweise, in Umgebungen operieren können, in der kein Zugang zu einem Netzwerk gewährleistet ist. (3) Die Sensoren zur Erfassung der Umgebung erzeugen zum Teil Datenmengen, für welche die Bandbreite heutiger Kommunikationstechnologien nicht ausreichend ist. Zwar werden ständig neue Kommunikationstechnologien entwickelt, die eine höhere Bandbreite zur Verfügung stellen. Zeitgleich werden jedoch auch die Sensoren stetig weiterentwickelt, beispielsweise um eine höhere Auflösung bereitzustellen, sodass auch die Datenmengen stetig ansteigen [4]. Folglich muss die Sensordatenverbreitung innerhalb der Roboter durchgeführt werden. Hierbei gibt es jedoch wiederum eine Reihe weiterer Einschränkungen, die den Einsatz leistungsfähiger Rechner verhindern. Zum einen ist der zur Verfügung stehende Platz sehr gering, zum anderen ist die Leistungsaufnahme beschränkt, um Akkulaufzeit zu maximieren und Wärmeentwicklung zu minimieren. Beides können leistungsfähige generische Prozessoren nicht erfüllen – üblicherweise nehmen sie zusammen mit der notwendigen Elektronik zu viel Platz ein, andererseits weisen sie einen hohen Energieverbrauch auf. Darüber hinaus erfüllen übliche Mehrkernprozessoren und Softwarearchitekturen typischerweise nicht die Anforderungen, die *harte Echtzeitsysteme* stellen.

Hardware-Beschleuniger

Eine Möglichkeit, um alle diese Anforderungen zu erfüllen, stellen jedoch *anwendungsspezifische Hardwarebeschleuniger* dar. Dabei handelt es sich um digitale Schaltkreise, die auf bestimmte Aufgaben zugeschnitten sind, um diese besonders effizient durchzuführen. Die Hardwarebeschleuniger übernehmen Aufgaben, die ansonsten üblicherweise als Software ausgeführt werden, um dadurch den eigentlichen Prozessor zu entlasten. Hardwarebeschleuniger erweisen sich oftmals hinsichtlich Rechengeschwindigkeit und elektrischer Leistungsaufnahme als wesentlich effizienter als generische Prozessoren. Besonders gut eignen sich diese Hardwarebeschleuniger für rechenintensive Aufgaben, die wiederholt auf einem Datenstrom ausgeführt werden sollen, also etwa für Algorithmen aus den Bereichen Digitale Signalverarbeitung, Bildverarbeitung und maschinelles Lernen. In der Vergangenheit erwies sich die Entwicklung solcher Hardwarebeschleuniger jedoch als äußerst komplex, zeit- und kostenintensiv. Ein aktuelles Forschungsthema ist am Robotics Innovation Center daher die Entwicklung des Frameworks reSPACE (reconfigurable Signal Processing and Classification Environment). Dieses soll die Entwicklung von Hardwarebeschleunigern für die Robotik wesentlich erleichtern. Die Algorithmen werden durch *modulare, wiederverwendbare Komponenten*, Knoten genannt, realisiert. Die Knoten beschreiben in einer generischen Form mittels Hardwarebeschreibungssprachen den Schaltkreis, der den jeweiligen Algorithmus

implementiert. Durch Auswahl bestimmter Parameter können die Knoten auf die spezielle Aufgabe zugeschnitten und optimiert werden, beispielsweise die Anzahl der zu verarbeitenden Kanäle und Samplingraten der Daten oder die erforderliche Rechengenauigkeit. Neben den spezifischen Knoten können generische Prozessorknoten verwendet werden, d. h. Knoten, welche die Funktionalität etwa eines Mikrocontrollers erfüllen und für allgemeinere Aufgaben verwendet werden, wie etwa die Überwachung und Verwaltung der Signalverarbeitung. Verschiedene Knoten können dann miteinander kombiniert werden, um komplexe Systeme, sogenannte *Designs*, zu erschaffen. Ein solches Design kann dann mithilfe von üblichen Synthesewerkzeugen in Netzlisten übersetzt werden, die es erlauben, FPGAs damit zu konfigurieren. Am Ende dieses Entwicklungsprozesses ist dadurch die gewünschte Anwendung als feste und effiziente Hardware schaltung realisiert.

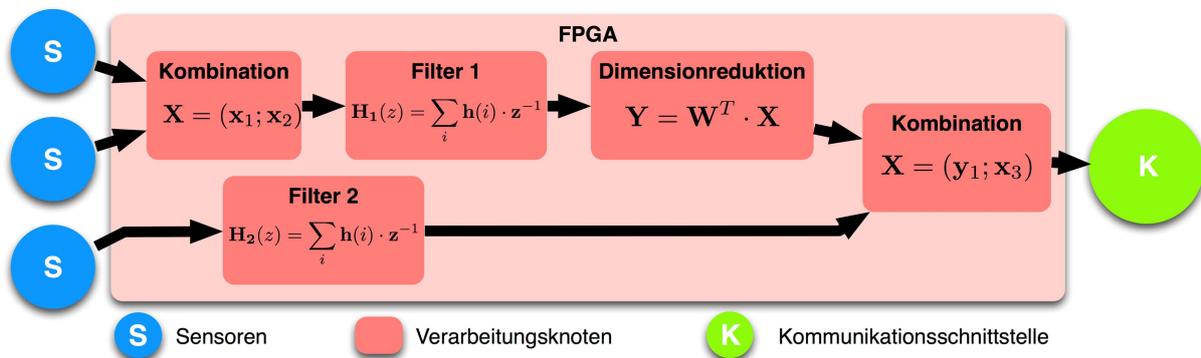


Bild 2: Datenflussbeschleuniger für die Robotik. Die Daten werden direkt von Knoten zu Knoten weitergeleitet und dabei transformiert. Hierbei können verschiedene Sensormodalitäten miteinander fusioniert werden. Die Ergebnisse werden an einen generischen Prozessor oder eine Kommunikationsschnittstelle übermittelt.

Datenfluss-Paradigma

Ein wesentliches Merkmal dieses Lösungsansatzes ist die Verwendung des Datenflussparadigmas. Klassische Prozessoren (sowohl die von-Neumann als auch die Harvard-Architektur) sind instruktionsgesteuert ein Strom von Anweisungen (das Programm) bestimmt, welche Operation mit einem bestimmten Datum als nächstes durchgeführt wird. Dies macht Rechner mit diesen Architekturen sehr generisch, jedoch weisen sie einige entscheidende Nachteile auf: (1) sie sind inhärent seriell – Parallelität bei Multiprozessor- oder Multikernprozessoren muss durch den Entwickler explizit implementiert werden, was sich häufig als kompliziert und fehleranfällig erweist [4]. (2) Sie besitzen einen Flaschenhals hinsichtlich des Speicherzugriffs, alle Daten sind im Hauptspeicher des Rechnersystems gespeichert und müssen vor der Verarbeitung erst zum Prozessorkern transportiert werden, was Zeit kostet. Dieses Problem wird durch Cache-Hierarchien zwar abgemildert, kann jedoch nicht vollständig gelöst werden. Diese Nachteile liegen bei Datenflussarchitekturen nicht vor. Hier *fließt* ein Strom von Daten durch die entsprechenden Knoten und wird dabei durch die in den Knoten implementierten Algorithmen transformiert. Sobald Daten für einen Knoten vorliegen, wird dieser aktiv und führt seine Operation auf den Daten aus. Liegen die Knoten als diskrete Schaltungen vor, können sie als *Pipeline* alle zur gleichen Zeit schon auf unterschiedlichen Daten des Datenstromes arbeiten, es liegt also schon auf einer hohen Ebene inhärenter Parallelismus vor. Darüber hinaus können alle Operatoren, die in den

Knoten implementiert werden, ebenfalls gleichzeitig aktiv sein. Neben dem Parallelismus minimiert eine solche Datenflussarchitektur auch das Problem des Speicherzugriffs: die Daten werden direkt von Knoten zu Knoten weitergeleitet, sodass ein Transfer der Daten hin und zurück vermieden werden kann. Ein potenzieller Nachteil ist hierbei jedoch die *Anwendungsspezifität* der Datenflussarchitektur: Die realisierte Struktur liegt als Hardware vor, die auf die aktuelle Anwendung zugeschnitten ist, für eine andere Anwendung eignet sich das jeweils aktuelle Design nicht. Dieses Problem kann jedoch durch zwei Strategien angegangen werden: Es werden auf Vorrat die zu erwartenden Knoten implementiert und der Datenstrom wird dynamisch umgeleitet oder das System wird je nach Anwendung *rekonfiguriert*.

Rekonfigurierbarkeit

FPGAs sind rekonfigurierbar, d. h. die Struktur der realisierten Schaltung kann je nach Anforderung geändert werden. Dadurch kann prinzipiell jede beliebige Anwendung realisiert werden, solange die Information über die gewünschte Schaltung vorliegt und genügend Platz auf dem FPGA vorhanden ist. Die Rekonfigurierbarkeit der Hardware erweist sich hierbei als wesentlicher Vorteil für die Industrie 4.0. Einerseits erweist sich die Verwendung von rekonfigurierbarer Hardware als kostengünstiger als die Verwendung von ASICs (Application Specific Integrated Circuit, d. h. *festverdrahtete Chips*, die genau auf eine Anwendung zugeschnitten sind), solange die Hardware nicht in extrem großen Stückzahlen gefertigt wird [5].

Andererseits erlaubt die Rekonfiguration, die gleiche Hardware für unterschiedliche Anwendungen zu nutzen, d. h. wenn für eine bestimmte Anwendung ein bestimmter Satz von Fertigkeiten des Roboters benötigt wird, können jeweils die passenden Hardwarebeschleuniger bereitgestellt werden. Ein weiterer Vorteil ist, dass stets aktuelle Methoden eingesetzt werden können, d. h. sobald neue, effizientere Algorithmen entwickelt werden, können diese beispielsweise in Form eines *Firmwareupdates* bereitgestellt werden.

Eingliederung der Roboter in Industrie-4.0 Architekturen

Das Rückgrat typischer Architekturen sowohl in der Industrie 4.0 und dem Internet der Dinge ist durch eine Reihe von Schichten realisiert, die sich aus drei wesentlichen Bausteinen zusammensetzen [6]: (1) Sensoren und Aktoren, (2) Embedded Processing sowie (3) Konnektivität und die Cloud. Wie in Bild 3 dargestellt, können hierbei die (1) und (2) in den Robotern selbst lokalisiert sein, die (3) direkt untereinander oder mit leistungsstarken Backen-Rechnern verbunden sind. Auf der Ebene der Roboter findet sämtliche Datenverarbeitung statt, die für aktuelle Aufgaben benötigt wird. Im Bereich der Cloud Robotik [7] können Daten in das Backend übertragen werden, um eine weitere Verarbeitung und Analyse zu erlauben. Ein weiterer Entwicklungsschritt wäre dann zukünftig das Bilden von Ad-hoc Clustern bestehend aus Robotern, die *direkt untereinander* Daten austauschen und Rechenkapazität *als Service* zur Verfügung stellen: Aktuell untätige Roboter rekonfigurieren ihre Hardware, um Rechenaufgaben anderer Roboter zu bearbeiten.

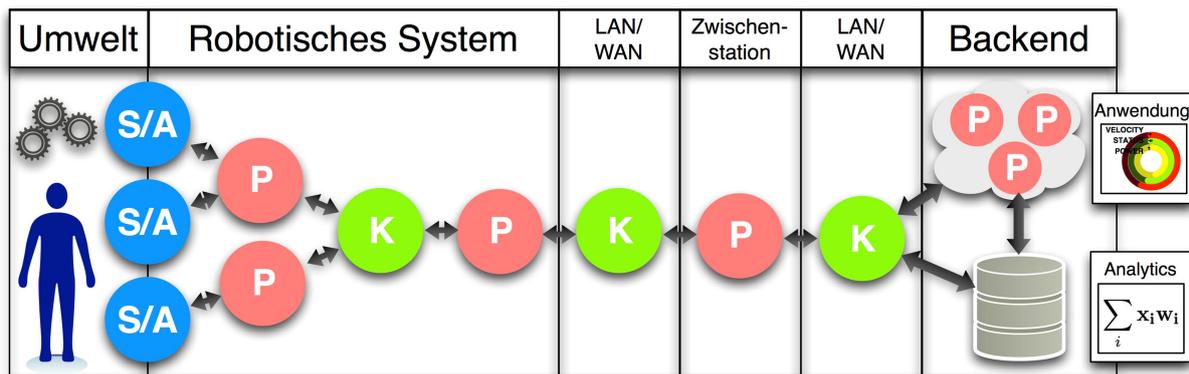


Bild 3: Architekturschema: Roboter mit Datenflussverarbeitung in einer Industrie 4.0 Architektur. In einem robotischen System findet schon eine dezentrale Verarbeitung statt. Daten können über weitere Zwischenstationen in ein Backend für z.B. übergeordnete Analysen übertragen werden.

Literatur

- [1] Hermann, M.; Pentek T.; Otto B.: Design Principles for Industrie 4.0 Scenarios: A Literature Review (Working Paper). Technische Universität Dortmund 2015. 
- [2] Kirchner, E. A.; de Gea Fernandez, J.; Kampmann, P.; Schröer, M.; Metzen, J. H.; Kirchner, F.: Intuitive Interaction with Robots–Technical Approaches and Challenges. In: Formal Modeling and Verification of Cyber-Physical Systems. Wiesbaden 2015, S. 224-248.
- [3] IBM: The Four V's of Big Data. URL: <http://www.ibmbigdatahub.com/infographic/four-vs-big-data>, abgerufen am: 9.11.2015
- [4] Asanovic, K.; Bodik, R.; Demmel, J.; Keaveny, T.; Keutzer, K.; Kubiawicz, J.; Yelick, K.: A view of the parallel computing landscape. In: Communications of the ACM 52 (2009) 10, S. 56-67.
- [5] Trimberger, S. M.: Three Ages of FPGAs: A Retrospective on the First Thirty Years of FPGA Technology. In: Proceedings of the IEEE 103 (2015) 3, S. 318-331.
- [6] Karimi, K.; Atkinson, G.: What the Internet of Things (IoT) needs to become a reality. White Paper, FreeScale and ARM. 2013.
- [7] Hu, G.; Tay, W. P.; Wen, Y.: Cloud robotics: architecture, challenges and applications. In: Network IEEE 26 (2012) 3, S. 21-28.

Schlüsselwörter:

Robotik, Cyber-Physische Systeme, Rechnerarchitektur, Maschinelles Lernen

Dieser Beitrag entstand im Rahmen der Projekte „Recupera-Reha“ und „TransTerra“, die vom Deutschen Zentrum für Luft und Raumfahrt unter den Kennzeichen 01IM14006A bzw. 50 RA 1301 gefördert werden.

Reconfigurable Dataflow Architectures in Robotics

Intelligent and autonomous robots are an essential part of the development of industry 4.0 solutions. They will act as a direct interaction partner to humans together in teams and

perform works that are much more complex than today's typical tasks for industrial robots. Such robots need to deal with a confusing and unpredictable environment and have to react to unforeseeable events. In order to capture this environment and to plan actions, real-time processing of complex sensor information is necessary. Conventional computer architectures appear to be insufficient for such kind of tasks. To solve this problem, hardware accelerators for robotics based on the dataflow paradigm are developed at the Robotics Innovation Center of the German Research Center for Artificial Intelligence.

Keywords:

robotics, cyber physical systems, computer architecture, machine learning