

Selbstlernende adaptive Robotersteuerung

Kontinuierliche Bewegungsplanung für Industrieroboter auf Basis von Sensordaten

Richard Meyes, Christian Scheiderer, Thomas Thiele und Tobias Meisen

In diesem Beitrag lesen Sie:

- ✓ wie künstliche neuronale Netze zum Steuern eines 6-Achs-Industrieroboters eingesetzt werden können,
- ✓ wie Roboter dazu befähigt werden können ihre Umgebung aktiv wahrzunehmen und
- ✓ warum diese Fähigkeit ein Reagieren auf sich dynamisch ändernde Umgebungszustände ermöglicht.

Der Prozess der Bewegungsplanung für Industrieroboter stellt eine komplexe Aufgabe dar, die häufig manuell von Domänenexperten durchgeführt wird und zu Lösungen führt, denen es an Flexibilität und Anpassungsfähigkeit zur Reaktion auf dynamische Umgebungseinflüsse mangelt. Dieses Problem lässt sich durch einen automatisierten Steuerungsagenten adressieren, der mit Hilfe eines künstlichen neuronalen Netzes und Reinforcement-Learning zur adaptiven und kontinuierlichen Bahnplanung befähigt wird. Der Agent lernt auf Basis seiner Wahrnehmung des Arbeitsraumes, welche durch Kamerabilder und ein Stromsignal gegeben ist, und ist hierdurch in der Lage, allgemeine Strategien für die Bewegung zu erlernen und umzusetzen. Neben der adaptiven Bewegungssteuerung ermöglicht dieser Ansatz dem Agenten auf sich ändernde Variationen der Umgebung und unkontrollierbare dynamische Umgebungseinflüsse zu reagieren und trägt dazu bei, die Zeit und Kosten von Ramp-Up-Prozessen zu reduzieren.

Einfluss dynamischer Umgebungsfaktoren

Industrieroboter sind eine der wichtigsten Komponenten für die industrielle Automation. Der weltweite Betriebsbestand an Industrierobotern hat sich in den letzten sieben Jahren fast verdoppelt und wird sich bis zum Jahre 2020 verdreifachen [1]. Trotz des weit verbreiteten Einsatzes von Industrierobotern auf der ganzen Welt ist die Erstinbetriebnahme des Roboters und die Programmierung der Roboterbewegung eine komplexe und kostspielige Aufgabe, die in der Regel manuell von Domänenexperten durchgeführt wird. Aufgrund des maßgeschneiderten Charakters der erforderlichen Bewegung für eine bestimmte Aufgabe

ist diese in der Regel fest einprogrammiert, um ein reproduzierbares Verhalten des Roboters innerhalb eines Produktionsprozesses zu gewährleisten. Solche fest einprogrammierten Bewegungen erfordern eine sorgfältige Kontrolle der Roboterumgebung, um sicherzustellen, dass keine dynamischen Faktoren die ausgeführte Roboterbewegung stören.

Problematisch ist der Ansatz bei Aufgaben, die aufgrund dynamischer Umgebungsbedingungen flexible und anpassungsfähige Bewegungen erfordern. Solche Aufgaben benötigen die Implementierung von Bewegungsplanungs- und Ausführungsalgorithmen, die in der Lage sind, sich dynamisch ändernde Umstände in der Umgebung zu berücksichtigen und gleichzeitig eine allgemeine Lösungsstrategie der Aufgabe zu erlernen. Einen derartig hohen Grad an Flexibilität in der Steuerungsmethodik lässt sich mit Hilfe von Methoden des maschinellen Lernens, insbesondere des Reinforcement Learnings prinzipiell erreichen. Reinforcement Learning ist eine spezifische Form des maschinellen Lernens, die es einem Agenten ermöglicht, seine Umgebung wahrzunehmen und mit ihr zu interagieren (Bild 1) [2]. Zu jedem Zeitschritt beobachtet der

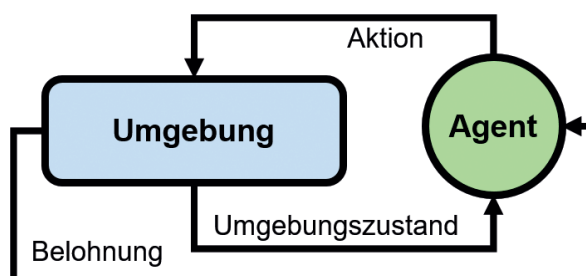


Bild 1: Schematische Darstellung des Reinforcement Learning Workflows

Richard Meyes M.Sc. ist wissenschaftlicher Mitarbeiter am Lehrstuhl für Informationsmanagement im Maschinenbau (IMA) an der RWTH Aachen University.

Christian Scheiderer M.Sc. ist wissenschaftlicher Mitarbeiter am Lehrstuhl für Informationsmanagement im Maschinenbau (IMA) an der RWTH Aachen University.

Dipl.-Ing. Thomas Thiele ist Berater der Nets, n' Clouds Consulting GmbH.

Prof. Dr.-Ing. Tobias Meisen ist Geschäftsführer des Lehrstuhls Informationsmanagement im Maschinenbau (IMA) an der RWTH Aachen University und besetzt die Juniorprofessur „Interoperabilität von Simulation“.

www.ima-zlw-ifu.rwth-aachen.de

Agent einen Zustand und führt eine Aktion aus, die eine entsprechende Belohnung liefert. Im Allgemeinen wird das Verhalten des Agenten durch eine Handlungsstrategie definiert, welche alle möglichen Zustände auf eine Wahrscheinlichkeitsverteilung über mögliche Aktionen abbildet und somit das Verhalten des Agenten bestimmt. Das Ziel von Reinforcement Learning ist es, eine Lösungsstrategie zu erlernen, die zu einer maximal möglichen Belohnung führt. Die Definition der Belohnungsfunktion beinhaltet implizit, welches Verhalten des Agenten erwünscht ist. So soll sie beispielsweise im Falle eines Fügeprozesses eine hohe Belohnung liefern, wenn dieser korrekt durchgeführt wurde und eine niedrige bzw. eine negative Belohnung, wenn der Prozess fehlschlägt und das zu fügende Bauteil beispielsweise während des Fügeprozesses beschädigt wird.

Demonstrationsszenario für adaptive Robotersteuerung

Als Demonstrationsszenario wurde ein Setup gewählt, in dem ein UR5-Industrieroboter [3] das klassische Spiel heißer Draht erlernt. Bild 2a zeigt experimentelle Setup und Bild 2b ein Beispielbild der Kamerabilder, die dem Agenten die Wahrnehmung seiner Umgebung erlauben und zum Lernen verwendet werden. Ein Kupferdraht von 150 cm Länge und einem Zentimeter Dicke befindet sich im Arbeitsbereich des Roboters und kann beliebig geformt werden. Der Roboter hält ein 3D-gedrucktes, maßgeschneidertes Werkzeug, welches bei Kontakt mit dem Draht einen Stromkreis schließt, wodurch Kollisionen mit dem Draht erkannt werden. Durch die Betrachtung eines zweidimensionalen Problems ergeben sich für den Roboter drei Freiheitsgrade für seine Bewegung. Aktionen werden somit durch eine Kombination von Längs- und Querbewegungen sowie einer Rotation beschrieben. Der Nutzen einer Aktion wird durch die jeweilig resultierende Belohnung beschrieben. Wenn

der Draht berührt wird oder wenn für eine bestimmte Anzahl von Zeitschritten kein signifikanter Fortschritt erreicht wurde, ist die Belohnung negativ, ein Fortschritt von fünf cm entlang des Drahtes ohne Kontakt wird mit einer positiven Belohnung bewertet.

Das Lernverhalten des Agenten wurde mit vier Drahtformen evaluiert, um seine Fähigkeit, das dargestellte Problem zu verallgemeinern sowie die Robustheit seiner erlernten Bewegungsstrategie zu untersuchen. Das erste Szenario erfordert, dass der Agent eine Strategie erlernt, mit welcher das Werkzeug ohne Berührung entlang des Drahts geführt wird. Das zweite Szenario verlangt, dass der Agent die erlernte Strategie auf eine neue Drahtform anwendet, ohne diese zuvor gesehen zu haben. Im dritten Szenario muss der Agent eine Passage überwinden, die eine präzisere Steuerung des Werkzeugs als in den vorangegangenen Szenarien erfordert. Schlussendlich testet das vierte Szenario die Robustheit des Lernprozesses unter starkem Hintergrundrauschen in den Kamerabildern. Hierzu wurde der schwarze Hintergrund durch einen HD-Plasma-Monitor ersetzt, auf dem während des Trainings zufällige YouTube-Videos abgespielt werden, um so ein dynamisches Prozessumfeld nachzustellen, wie es beispielsweise in einem realen Montageszenario aus dem industriellen Umfeld vorliegt.

Flexibilität unter variablen Umgebungsbedingungen

Unter den angeführten Rahmenbedingungen konnte der Algorithmus das erste Szenario nach 23 Minuten fehlerfrei und robust bewältigen. Im Trainingsprozess wurde der Draht insgesamt 650 Mal berührt, was sich in den im Bild 3 dargestellten benötigten Trainingsiterationen widerspiegelt. Konfrontiert mit der zweiten Drahtform war der Agent mit 13 Minuten Trainingszeit deutlich schneller. Auch die Anzahl an Trainingsiterationen reduzierte sich durch den bereits vortrainierten Agenten auf 150. Im dritten Szenario zeigte

Literatur

- [1] International Federation of Robotics (IFR), Executive Summary World Robotics 2017 Industrial Robots (2017).
- [2] Andrew G. Barto, Richard S. Sutton, Reinforcement Learning: An Introduction, Second Edition (2017).
- [3] Universal Robots, UR5 Technical specifications (2016).
- [4] Richard Meyes, Christian Scheiderer, Tobias Meisen, Continuous Motion Planning for Industrial Robots based on Direct Sensory Input (2018), Procedia CIRP (to be published)

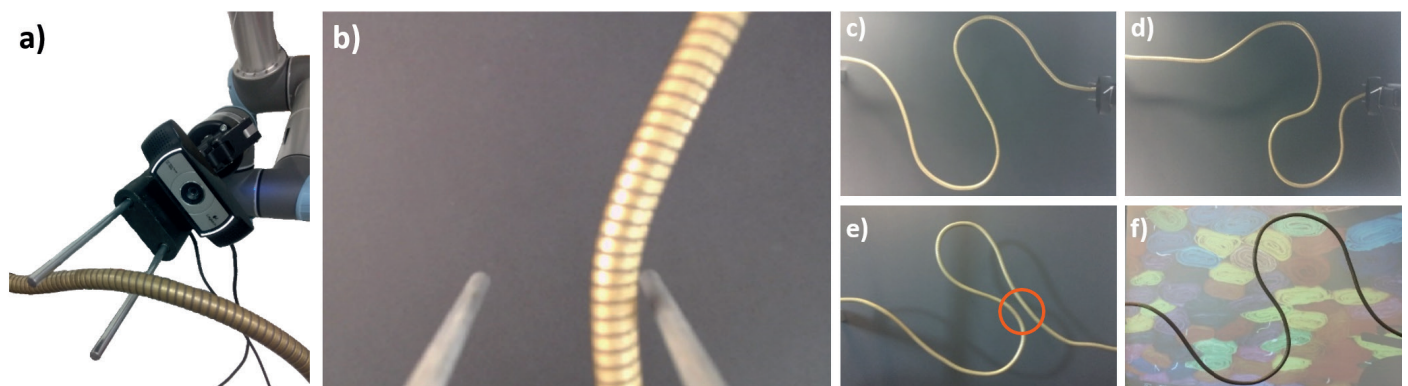


Bild 2 a) Ausschnitt des Robotergräfers mit Kamera, b) Sichtfeld des Agenten, c-f) Die vier Lernszenarien mit ansteigendem Schwierigkeitsgrad

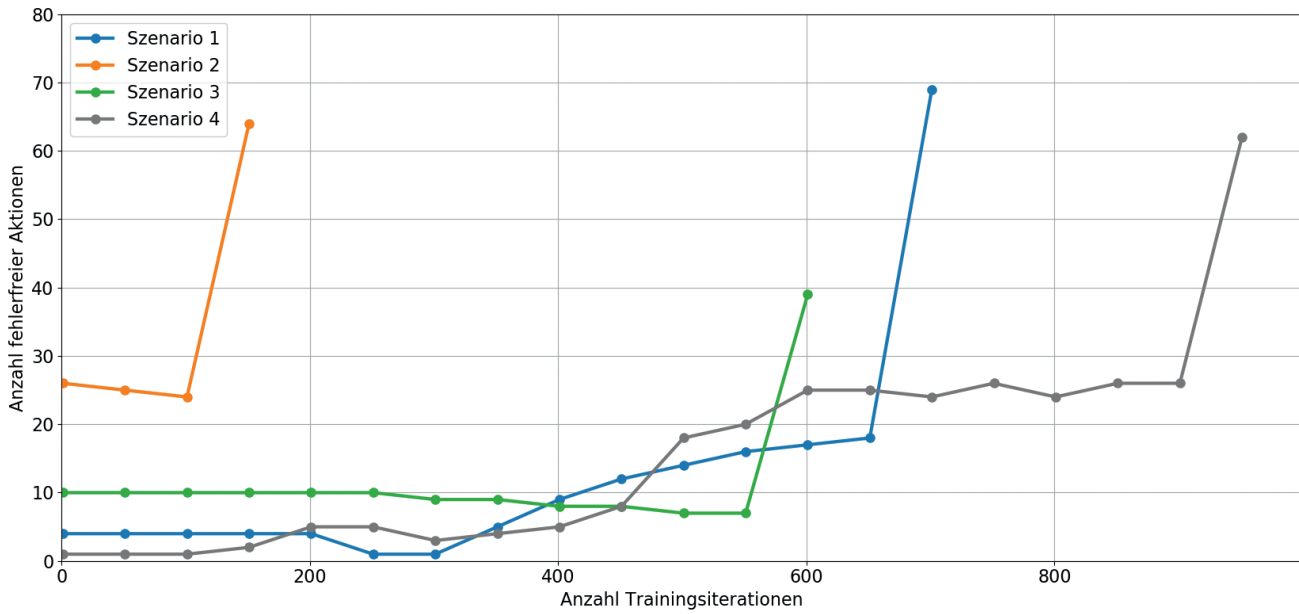


Bild 3: Lernfortschritt des Agenten in den vier Szenarien

der Agent, dass selbst höhere Anforderungen an die Präzision der Bewegungen gemeistert werden können. Die Zunahme der Komplexität hatte jedoch eine erhöhte Trainingszeit (35 Minuten) und Anzahl an Fehlversuchen (600) zur Folge. Im letzten Szenario schnitt der Agent im Vergleich zu den anderen Szenarien am Schlechtesten ab (61 Minuten und 950 Fehlversuche). Jedoch wurde auch in diesem dynamischen Szenario eine Lösungsstrategie erlernt, die zu einer erfolgreichen Bewältigung der Aufgabe ohne Eingriffe durch den Menschen führte.

Die Experimente zeigen, dass der gewählte Ansatz innerhalb kurzer Zeit zu einer Bewegungsführung in der realen Welt führt. Die hierfür gewählten Referenzszenarien bilden eine steigende Komplexität hinsichtlich der Bewegungsführung ab, um die Robustheit des Verfahrens bei gleichzeitig steigenden Anforderungen an die Genauigkeit der Bewegung zu demonstrieren. Detailliertere Informationen bezüglich des verwendeten Trainingsalgorithmus und der Topologie des neuronalen Netzes finden sich in [4].

Übertragbarkeit auf industrielle Anwendungen

Die Ergebnisse zeigen, dass mittels Reinforcement Learning trainierte, künstliche neuronale Netze in der Lage sind, Bewegungsplanungsaufgaben für 6-Achs Industrieroboter zu erlernen und auf unvorhergesehene dynamische Einflüsse im Arbeitsbereich des Roboters zu reagieren. Aufbauend auf den gezeigten Ergebnissen besteht der nächste Schritt in der Übertragung der verwendeten Methodik auf industrielle Anwendungen wie zum Beispiel das Setzen von Kleberaupen oder das Fügen von biegeschlaffen Bauteilen. Um in derartigen industriellen Umfeldern die Anzahl an Fehldurchläufen während des Lernprozesses zu reduzieren, soll der Lernprozess in einer Simulationsumgebung stattfinden, die dem lernenden Agenten erlaubt Fehler zu machen, ohne dass es zu Schäden in der realen Umgebung kommt. Das in der Simulationsumgebung erlernte Verhalten kann dann als solide Initiallösung in die Realität übertragen und dort durch zusätzliches Lernen verfeinert werden.

Schlüsselwörter:

Kontinuierliche Bewegungen, Industrieroboter, Reinforcement Learning, künstliche neuronale Netze, Kamerabilder, selbstlernende Systeme, dynamische Prozessbedingungen, adaptive Regelung

Danksagung

Die hier vorgestellten Ansätze werden von der Deutschen Forschungsgemeinschaft (DFG) im Rahmen des Exzellenzclusters „Integrative Produktionstechnik für Hochlohnländer“ an der RWTH Aachen unterstützt.

Self-Learning Adaptive Robot Control

The process of motion planning for industrial robots is a complex task that is often performed manually by domain experts and leads to solutions that lack flexibility and adaptability with respect to dynamic environmental influences. This problem can be addressed by an automated control agent, which enables adaptive and continuous path planning by means of an artificial neural network and reinforcement learning. The agent learns on the basis of its perception of the working space, which is given by camera images and a current signal, and is thus able to learn and implement general strategies for the movement. In addition to adaptive motion control, this approach enables the agent to respond to changing variations in the environment and uncontrollable dynamic environmental influences, helping to reduce the time and cost of ramp-up processes.

Keywords:

Continuous Motion, Industrial Robots, Reinforcement Learning, Artificial Neural Networks, Camera Images, Self-Learning Systems, Dynamic Process Conditions, Adaptive Control