Potenziale von Reinforcement Learning für die Produktion

Marco Huber, Tobias Nagel, Raphael Lamprecht und Florian Eiling, Fraunhofer IPA, Stuttgart

Reinforcement Learning (RL) konnte bereits publikumswirksam in Video- und Strategiespielen beeindruckende Erfolge erzielen [1]. Diese Grundlagenforschung schafft die Grundlagen, dass RL für reale Entscheidungsprobleme in der Produktion nutzbar wird. Beispiele hierfür sind: Wie erhält ein Roboter mehr Intelligenz, um Aufgaben selbstständiger und ohne aufwendige Programmierung durchzuführen? In welcher Reihenfolge müssen Aufträge in einer Produktion abgearbeitet werden, um eine optimale Termintreue zu erhalten? Der Beitrag gibt eine Einführung in die Arbeitsweise des RL, sowie dessen bevorzugte Einsatzgebiete und beschreibt Anwendungsbeispiele aus dem produzierenden Alltag. Das präsentierte Überblickswissen über die aktuelle Forschung soll diesen Teilbereich der Künstlichen Intelligenz einem breiteren Interessentenkreis zugänglich machen. Übergeordnetes Ziel der beschriebenen Methoden ist, die Wertschöpfung am Wirtschaftsstandort Deutschland kontinuierlich zu steigern.

Potentials of Reinforcement Learning for Production

Reinforcement learning (RL) can be more and more used for real-world decision problems in production. The article gives an introduction into the functionalities of RL as well as its preferred areas of application. It further describes project examples from everyday production. The presented knowledge of current research is intended to make this sub-area of artificial intelligence accessible to a broader audience and to increase the added value in production.

Keywords:
reinforcement learning, autonomous production and job control

In den letzten Jahren wurden die Entwicklung und Anwendung von Methoden der Künstlichen Intelligenz (KI) stark vorangetrieben, sodass sie auch im produzierenden Umfeld Einsatz gefunden haben. Diese Methoden beruhen meist auf sogenannten überwachten Lernverfahren, die Bild- oder Sensordaten mit dem Maschinenzustand oder der Produktgüte in Beziehung setzen. Beispielhaft dafür sind optische Qualitätsprüfungen, die die Güte eines Produkts bewerten, oder Verfahren der vorausschauenden Instandhaltung, welche die frühzeitige Prognose von anstehenden Wartungen oder Maschinenausfällen ermöglichen. Notwendig für ein Training dieser KI-Verfahren sind in der Regel neben den Bild- und Sensordaten auch erhobene und annotierte Zielgrößen.

Unüberwachte Lernverfahren kommen ohne eine konkrete Annotation aus und werden häufig für die Anomalieerkennung verwendet. Dazu kann zum Beispiel ein neuronales Netz mit relevanten Messdaten der Anlage im Normalbetrieb trainiert werden. Reproduziert dabei das Netz die Messdaten, so spricht man von einem Autoencoder [2]. Wenn nach dem Training eine Anomalie auftritt, so gibt die Anlage ein abweichendes Verhalten wieder und der Autoencoder sollte eine große Abweichung zwischen den Messdaten und der Netzausgabe liefern. Weil aber die Zielgröße bei komplexen Aufgaben der Planung und Entscheidung häufig unbekannt ist, können diese Aufgaben nicht mit den ‚klassischen' Verfahren des maschinellen Lernens (ML) gelöst werden. Ein Beispiel dafür ist das Problem einer optimalen Auftragssteuerung, die eine möglichst hohe Termintreue erreicht. Es ist unklar, welche Auftragsreihenfolge zum gewünschten Ziel führt. Ein weiteres Beispielszenario stellt das Griff-in-die-Kiste-Problem dar, bei dem ein Roboter einzelne, chaotisch gelagerte Objekte in einer Kiste greifen und diese geordnet ablegen soll. Da ständig wechselnde Objektformen ein jeweils angepasstes Greifen erfordern, ist die notwendige Greifstrategie nicht bekannt. In diesem wie im oben dargestellten Fall der Auftragssteuerung können keine mathematischen Optimierungsverfahren genutzt werden. Aber es gibt dennoch eine geeignete ML-Methode: In den beschriebenen Fällen spielen die bestärkenden Verfahren, das sogenannte Reinforcement Learning (RL), ihre Mehrwerte aus.

Funktionsweise des Reinforcement Learning

RL-Methoden funktionieren über einen belohnungsorientierten Algorithmus: Ein Agent, der sich in einem Zustand S befindet, kann eine Aktion A ausführen. Diese wird mit dem Ziel gewählt, eine Belohnung R (engl. Reward) zu maximieren (Bild 1). Genaueres ist [3] zu entnehmen. Konkret angewandt werden kann

Prof. Dr.-Ing. Marco Huber ist stv. Leiter des Instituts für Industrielle Fertigung und Fabrikbetrieb IFF der Universität Stuttgart und Leiter der Abteilung Bild- und Signalverarbeitung sowie des Zentrums für Cyber Cognitive Intelligence (CCI) am Fraunhofer-Institut für Produktionstechnik und Automatisierung IPA.

Tobias Nagel, M. Sc. ist Mitarbeiter am CCI am Fraunhofer-Institut für Produktionstechnik und Automatisierung IPA. Seine Forschungsschwerpunkte umfassen Regelungstechnik verbunden mit Verfahren der Künstlichen Intelligenz.

Raphael Lamprecht, M. Sc. ist Mitarbeiter am CCI am Fraunhofer-Institut für Produktionstechnik und Automatisierung IPA. In seiner Forschung beschäftigt er sich mit dem Einsatz von Methoden der Künstlichen Intelligenz, um Planungs- und Steuerungsprobleme in der Produktion zu lösen.

Florian Eiling, M. Sc. ist Mitarbeiter in der Gruppe Kognitive Produktionssysteme am Institut für Industrielle Fertigung und Fabrikbetrieb IFF der Universität Stuttgart. Sein Forschungsschwerpunkt liegt auf der Anwendung von modellbasiertem Reinforcement Learning zur Steuerung von Produktionsprozessen.

marco.huber@ipa.fraunhofer.de
www.ipa.fraunhofer.de/ki

https://doi.org/10.30844/I40M_21-2_S25-29

Bild 1: Funktionsweise des Reinforcement Learning. Quelle: Fraunhofer IPA.

Derartiges beispielsweise beim oben eingeführten Griff-in-die-Kiste-Problem. Hierbei entspricht der Roboter dem Agenten, sodass der Zustand unter anderem durch die Aufnahme von Bildern aus der Sicht des Roboters festgelegt werden kann. Mögliche Aktionen sind das Bewegen des Roboterarms und -greifers in verschiedenen Posen. Das erfolgreiche Greifen eines Objekts entspricht dann einer hohen Belohnung.

Meistens werden hier noch sogenannte modellfreie RL-Algorithmen eingesetzt, das heißt es ist kein Dynamikmodell der Umwelt des Agenten für das Training notwendig. Stattdessen lernt der Algorithmus eine Handlungsstrategie direkt, also mit welcher Aktion, bezogen auf einen Zustand, mit der höchsten Belohnung zu rechnen ist. Das bedeutet konkret für den Griff-in-die-Kiste: Liegt ein Bauteil an Position x, plant der Roboter den Pfad sowie das Greifen derart, dass er das Bauteil sicher aufnehmen kann.

Eine hierfür häufig verwendete Algorithmenklasse sind die sogenannten Policy-Optimization-Algorithmen. Diese starten zumeist mit einer randomisierten Strategie und verbessern diese iterativ, basierend auf den erhaltenen Belohnungen [4, 5]. Ein weiterer Vertreter dieser modellfreien RL-Algorithmen ist das Q-Learning. Beim Q-Learning wird für jede zulässige Aktion in allen Zuständen des Systems ein Wert (Q-value) evaluiert, der Rückschlüsse darüber zulässt, wie viel Belohnung in einem Zustand zu erwarten ist. Der Agent leitet daraus die Strategie ab, indem immer die Aktion ausgewählt wird, die den Q-value maximiert [3].

Um erfolgreich zu trainieren bzw. zu lernen, benötigt das RL-Verfahren Rückmeldungen aus seiner Umwelt, wie gut die gerade durchgeführte Aktion war. Dieses Training könnte einerseits in einer realen Umgebung durchgeführt werden, was jedoch nicht optimal ist: Es führt zu einer Kapazitätsblockade und erhöhtem Verschleiß. Diese sind oftmals nicht vertretbar, weil modellfreie Algorithmen sehr viele Iterationen benötigen, um ein akzeptables Ergebnis zu erreichen.

Eine Simulation hat den Vorteil, dass das Training schneller als in Echtzeit durchgeführt werden kann. Allerdings ist die Erstellung einer Simulation häufig mit erheblichem Zeitaufwand verbunden und spiegelt die Realität nur begrenzt wider. Und selbst bei der Verwendung von modernen Physiksimulationen können nicht alle realen physikalischen Effekte berücksichtigt werden. Deswegen kommt es zu einem Leistungsverlust, wenn ein in der Simulation gelernter Algorithmus in die reale Welt übertragen wird. In diesem Zusammenhang spricht man vom Reality- oder Sim2Real-Gap. Der Leistungsverlust kann aber durch Techniken wie Domain Randomization stark abgemildert werden. Bei der Verwendung von Domain Randomization werden die Physikparameter der Simulation zufällig gewählt und regelmäßig verändert. Dadurch wird der Algorithmus robuster gegenüber Ungenauigkeiten in der Simulation und kann somit besser auf reale Hardware übertragen werden [6, 7].

Abhilfe verschaffen sogenannte modellbasierte RL-Algorithmen. Der Programmierer kann hier entweder über bereits vorhandene Vergangenheitsdaten oder über durchgeführte Messungen ein Dynamikmodell der Umgebung erstellen, das online aktualisiert wird. Dieses Dynamikmodell kann häufig mit überwachten Lernmethoden erzeugt werden und wird dazu genutzt, um mit klassischer, modellbasierter Prädiktivregelung [8] herauszufinden, welche auszuführende Aktion die höchste Belohnung entgegenbringt. Nachdem der Agent die Aktion mit der am höchsten zu erwartenden Belohnung durchgeführt hat, werden die nun neu erhaltenen Daten direkt verwendet, um das erstellte Dynamikmodell online zu aktualisieren (Bild 2). Sollte sich das Systemverhalten während des Betriebs verändern, zum Beispiel, wenn in einer Kiste neue

Bauteile hinzugefügt werden oder sich die Greifer abnutzen, wird das Dynamikmodell im Betrieb angepasst, sobald Prädiktion und Messung voneinander abweichen.

Vorteilhaft ist, dass modellbasierte RL-Algorithmen häufig mit deutlich geringeren Datenmengen auskommen und somit schneller auf ein gutes und nutzbares Ergebnis kommen. Im Folgenden werden vier verschiedene Beispielanwendungen vorgestellt, bei denen Reinforcement Learning bereits heute erfolgreich eingesetzt wird.

Auftrags- und Produktionssteuerung

In der Produktionssteuerung existiert eine Vielzahl an Teilproblemen. Diese unterscheiden sich sowohl hinsichtlich des Umfangs der betrachteten Teilaspekte des Produktionssystems als auch durch die Ein- und Ausgangsgrößen des Planungsproblems. Wichtige Planungsprobleme sind die Auftragsreihenfolge- und Maschinenbelegungssteuerung. Für diese Steuerung werden heute hauptsächlich Dispatching-Heuristiken wie beispielsweise die „First-in-First-Out"-Logik eingesetzt [9]. Heuristiken lassen sich einfach implementieren, sind leicht verständlich und können auf beliebige Produktionssysteme angewandt werden.

Hinsichtlich der Lösungsqualität sind Heuristiken jedoch oftmals nicht optimal, weshalb Unternehmen auch auf mathematische Optimierung zurückgreifen möchten, um Planungsprobleme zu lösen. Im Gegensatz zu Heuristiken liefern diese oftmals bessere Ergebnisse. Allerdings steigt die benötigte Rechenzeit, die für die Lösung des Optimierungsproblems benötigt wird, exponentiell mit der Problemgröße. Deshalb sind mathematische Optimierungen bisher nur für vergleichsweise kleine Systeme geeignet. Um aber auch bei komplexen Produktionssystemen eine gute Lösungsqualität zu ermöglichen, arbeiten Wissenschaftler des Fraunhofer-Instituts für Produktionstechnik und Automatisierung IPA im Forschungsprojekt „RESYST" aktuell daran, RL für die autonome Produktionssteuerung zu nutzen (Bild 3).

Um den RL-Agenten zu trainieren, ist eine Umgebung nötig, mit der der Agent interagieren kann. Dies ist im realen Produktionssystem nicht zielführend, denn die zunächst schlechten Entscheidungen des Agenten würden betriebliche Abläufe direkt beeinflussen. Daher ist für dieses Training eine Simulationsumgebung erforderlich. Hierfür wird oftmals ein sogenanntes „ereignisorientiertes Simulationsmodell" verwendet, das einen digitalen Zwilling der zu steuernden Produktion darstellt. Aktuell wird häufig das „Deep Q-Learning (DQN)" eingesetzt, ein modellfreier Algorithmus, der eine Variante des Q-Learning darstellt [10]. Auch in „RESYST" erlernt ein DQN-Agent während des Trainings eine Strategie, um Produktionsaufträge Maschinen zuzuordnen und so die Maschinenbelegungssteuerung zu realisieren. Im nächsten Schritt kann diese Strategie dann außerhalb der Simulation in der realen Fabrik eingesetzt werden, um das Produktionssystem zu steuern.

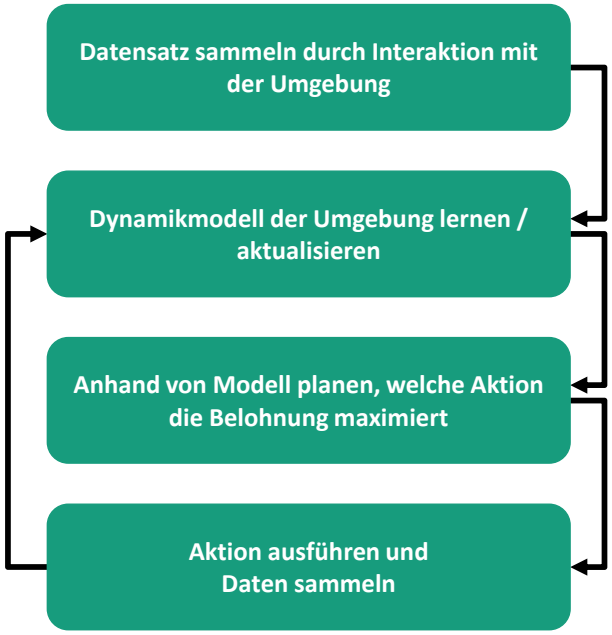

Bild 2: Schema des modellbasierten Reinforcement Learning. Quelle: Fraunhofer IPA.

Um den Trainingsprozess weiter zu beschleunigen, kann auch ein Vor-Training durchgeführt werden, welches auf historischen Produktionsdaten basiert und somit die bisherige Entscheidungsfindung abbildet. Auf diese Weise erzeugt der RL-Algorithmus zu Beginn keine zufälligen Kombinationen. Das weitere Training dient anschließend dazu, das bisherige Verfahren zu optimieren. Die erlernte Strategie hängt dabei davon ab, wie die Belohnungsfunktion ausgestaltet ist. Sie kann entsprechend den relevanten Zielgrößen des Produktionssystems beschrieben werden. So lassen sich konkurrierende Zielgrößen wie geringe Bestände, hohe Auslastung oder hohe Termintreue in der Zielfunktion gewichten.

Neben der Ausgestaltung der Belohnungsfunktion beeinflusst die Auswahl des Algorithmus maßgeblich das Lernverhalten. Struktur und Größe des Zustands- und Aktionsraums bedingen unter anderem diese Auswahl. Der Zustandsraum ist durch die Menge aller möglichen Zustände beschrieben, in denen sich

https://doi.org/10.30844/I40M_21-2_S25-29

Literatur

[1] D. Silver, J. Schrittwieser, K. Simonyan, I. Antonoglou, A. Huang, A. Guez, T. Hubert, L. Baker, M. Lai, A. Bolton, Y. Chen, T. Lillicrap, F. Hui, L. Sifre, G. van den Driessche, T. Graepel und D. Hassabis, „Mastering the game of Go without human knokwledge," Nature 550, Oktober 2017.

[2] A. Borghesi, A. Bartolini, M. Lombardi, M. Milano und L. Benini, „Anomaly Detection Using Autoencoders in High Performance Computing Systems," in The Thirty-First AAAI Conference on Innovative Applications of Artificial Intelligence, 2019.

[3] R. S. Sutton und A. G. Barto, Reinforcement learning: An introduction, MIT press, 2018.

[4] T. Haarnoja, A. Zhou, K. Hartikainen, G. Tucker, S. Ha, J. Tan, V. Kumar, H. Zhu, A. Gupta, P. Abbeel und S. Levine, „Soft Actor-Critic Algorithms and Applications," 2018.

Bild 3: Reinforcement Learning ermöglicht, den Materialtransport im Produktionssystem optimal zu steuern. Quelle: Fraunhofer IPA/IFF Universität Stuttgart/Foto: Rainer Bez.

das Produktionssystem befinden kann. Der Aktionsraum hingegen beschreibt, welche Aktionen der Agent ausführen kann, um mit dem Produktionssystem zu interagieren. In weiteren Forschungsarbeiten soll untersucht werden, wie Modellwissen zur Planung in den Trainingsprozess integriert werden kann. Dies ermöglicht den Einsatz von Suchalgorithmen wie beispielsweise „Monte Carlo Tree Search" (MCTS) [11] und kann insbesondere bei großen Aktions- und Zustandsräumen helfen, den RL-Agenten effizienter zu machen.

Optimierte Rüstreihenfolge

Eine verwandte Problemstellung, bei der RL in der Produktion gewinnbringend eingesetzt werden kann, liegt in der Reihenfolgebildung von Rüstaufträgen. Bei rüstaufwendigen Produktionsabläufen stellt sich die Frage, in welcher Reihenfolge die Produktionsaufträge in die Produktion eingebucht werden sollen, damit der durch die Rüstung bedingte Produktionsstillstand möglichst gering ist. Auch diese Problemstellung lässt sich als RL-Aufgabe formulieren: Der Zustand wird aus der aktuellen Auftragsreihenfolge zusammengesetzt, während der Aktionsraum aus dem Hinzufügen oder Entfernen von Aufträgen an bestimmten Stellen besteht. Anschließend kann der Agent die Reihenfolge der Aufträge unter Berücksichtigung der relevanten Zielgrößen bilden. Die Funktionstüchtigkeit dieses Ansatzes konnte bereits erfolgreich in einem produzierenden Unternehmen bewiesen werden, indem die Rüstdaten mehrerer Hunderter Aufträge analysiert und verarbeitet wurden.

Ähnlich wie im vorigen Abschnitt ist es zunächst nicht ratsam, das komplette Training in der echten Produktion durchzuführen. Zwar könnte man auf einen digitalen Zwilling ausweichen. Dies stellt jedoch einen erheblichen Mehraufwand dar. Eine weitere Option besteht darin, das RL-Verfahren zunächst als Assistenzsystem zu implementieren. Dieses schlägt eine Auftragsreihenfolge vor, die ein Mitarbeiter dank seiner Erfahrung einfach annehmen oder ablehnen kann. Sollte der Mitarbeiter den RL-Vorschlag ablehnen, wird der Belohnungswert verkleinert in das Verfahren zurückgeführt, was einen entsprechenden Lerneffekt zur Folge hat. Bei Annahme des Vorschlags ist die Belohnung größer. Auf diese Weise erlernt das RL-System, iterativ bessere Vorschläge zu generieren, mit dem Ziel, dass der Mitarbeiter nach einer entsprechenden Zeit alle Vorschläge annimmt.

Instandhaltungsplanung

Ein weiteres Anwendungsgebiet, das im KI-Fortschrittszentrum Lernende Systeme und Kognitive Robotik am Fraunhofer IPA erforscht wird, ist der Einsatz von RL zum Erlernen von Instandhaltungsstrategien, um die vorausschauende Wartung oder „Predictive Maintenance" umzusetzen. Bisher werden Wartungs- und Instandhaltungstätigkeiten oftmals anhand starrer Wartungspläne durchgeführt, was dazu führt, dass Anlagen teilweise zu früh oder zu spät gewartet werden. Dadurch wird die Produktion häufig unnötigerweise unterbrochen oder produktionsfreie Zeiten nicht für Instandhaltungstätigkeiten genutzt, was wiederum zu ungeplanten Anlageausfällen führen kann.

RL kann dabei helfen, auf Basis des aktuellen Zustands des Produktionssystems Wartungs- und Instandhaltungstätigkeiten sinnvoller zu terminieren. Auch hier wird das Produktionssystem zum Training in einer Simulation dargestellt. In dieser lassen sich stochastische Prozesse abbilden, wie beispielsweise der Ausfall von Anlagen durch Verschleiß oder variierende Bearbeitungszeiten für Wartungs- und Instandhaltungstätigkeiten.

Robotik- und Prozessregelung

Im Forschungsprojekt „rob-aKademl" erforschen IPA-Wissenschaftler gemeinsam mit dem Institut für Industrielle Fertigung und Fabrikbetrieb IFF der Universität Stuttgart und vier weiteren Projektpartnern den Einsatz von RL zum automatisierten Lernen von Steuerungsalgorithmen für Montageanwendungen. Bisher stellen Montageprozesse klassische Roboterprogrammiermethoden vor große Herausforderungen. Maßgeblich dafür sind vor allem die Komplexität und große Variantenvielfalt der Prozesse. Außerdem benötigen klassische Roboterprogrammiermethoden

einen hohen Zeit- und Personalaufwand. Selbst mit modernen, intuitiven Programmiermethoden wäre der Aufwand für immer stärker personalisierte Produkte noch derart hoch, dass er den Aufwand der manuellen Produktion übersteigen würde.

Der Ansatz von rob-aKademl möchte dieses Problem durch die Verwendung von RL zum automatisierten Lernen von Steuerungsalgorithmen für komplexe Montageprozesse in kleinen Losgrößen lösen. Dafür wird die vom IPA entwickelte, skill-basierte und kraftgeregelte Software pitasc mit modernen RL-Methoden kombiniert. Es wird ein hybrider RL-Algorithmus verwendet, um die von pitasc zur Verfügung gestellten Fähigkeiten, wie z. B. eine einfache Bewegung des Roboterarms an eine definierte Stelle, zu kontrollieren und zu einem Gesamtprozess zusammenzufügen. Der RL-Algorithmus muss somit nicht die einzelnen Fähigkeiten neu lernen, sondern kann auf bereits bestehende Fähigkeiten zurückgreifen. Dieser hybride Ansatz erhöht die Dateneffizienz, die Übertragbarkeit auf ähnliche Problemstellungen und die Robustheit der RL-Algorithmen deutlich.

Bild 4: Die Programmierung von Robotern für Montageaufgaben soll mit Reinforcement Learning deutlich einfacher möglich werden. Quelle: Fraunhofer IPA/Foto: Rainer Bez.

Eine weitere Stärke des rob-aKademl-Ansatzes liegt im Training in der Simulation. Wie oben bereits ausgeführt, würde ein direktes Training auf dem Roboter zu hohem Hardwareverschleiß, Ausschuss und einem Stillstand der Produktionssysteme führen. Außerdem wäre das Training aufgrund einer beschränkten Bewegungsgeschwindigkeit des Roboterarms entsprechend langsam. Um dieses Problem zu umgehen, findet das Lernen bei rob-aKademl rein in der Simulation statt. Eine moderne Simulationsumgebung kann die Prozesse mit hoher Genauigkeit abbilden und durch den Parallelbetrieb vieler Instanzen die große Menge benötigter Daten erzeugen. Nach dem Abschluss des Trainings werden die gelernten Algorithmen auf das reale System übertragen. Um das System robuster zu machen, werden mittels Domain Randomization viele zufällige Szenarien mit variierenden physikalischen Parametern erzeugt [12].

Weitere Informationen:

Informations- und Fördermöglichkeiten: Unternehmen aller Branchen und Größen können sich mit allen ML-Fragen und Umsetzungsideen an das Zentrum für Cyber Cognitive Intelligence (CCI) am Fraunhofer IPA wenden. Zudem bietet das IPA zusammen mit dem Fraunhofer-Institut für Arbeitswirtschaft und Organisation IAO verschiedene Formen der Zusammenarbeit im KI-Fortschrittszentrum Lernende Systeme und Kognitive Robotik. Dieses ist Teil des KI-Forschungsverbundes Cyber Valley und unterstützt im Besonderen beim Transfer von der Grundlagenforschung in die Anwendung.

Das CCI wird vom Ministerium für Wirtschaft, Arbeit und Wohnungsbau des Landes Baden-Württemberg unter dem Förderkennzeichen 017-192996 gefördert. Das KI-Fortschrittszentrum im Forschungsverbund Cyber Valley wird ebenfalls vom Ministerium für Wirtschaft, Arbeit und Wohnungsbau des Landes Baden-Württemberg unter dem Förderkennzeichen 036-170017 gefördert. Das Projekt „rob-aKademl" mit dem Förderkennzeichen 01IS20009 erhält finanzielle Mittel vom Bundesministerium für Bildung und Forschung. Das Projekt REYST wird vom Bundesministerium für Bildung und Forschung gefördert.

Schlüsselwörter:
Reinforcement Learning, Autonome Produktions- und Prozessteuerung, Robotik

[5] J. Schulman, F. Wolski, P. Dhariwal, A. Radford und O. Klimov, „Proximal Policy Optimization Algorithms," arXiv Preprint, 2017.
[6] X. Peng, W. Andrychowicz, W. Zaremba und P. Abbeel, „Sim-to-Real Transfer of Robotic Control with Dynamics Randomization," 2017.
[7] Y. Chebotar, A. Handa, V. Makoviychuk, M. Macklin, J. Issac, N. Ratliff und D. Fox, „Closing the Sim-To-Real Loop: Adapting Simulation, Randomization with Real World Experience," 2018.
[8] R. Dittmar und B.-M. Pfeiffer, Modellbasierte prädiktive Regelung: Eine Einführung für Ingenieure, Walter de Gruyter, 2009.
[9] B. Waschneck, Autonome Entscheidungsfindung in der Produktionssteuerung komplexer Werkstattfertigungen, Stuttgart: 2020.
[10] T. Altenmüller, T. Stüker, B. Waschneck, A. Kuhnle und G. Lanza, „Reinforcement learning for an intelligent and autonomous production control of complex job-shops under time constraints," Production Engineering 14, 2020.
[11] D. Silver und J. Veness, „Monte-Carlo Planning in Large POMDPs," (NIPS) Advances in Neural Information Processing Systems, 2010.
[12] M. El-Shamouty, K. Kleeberger, A. Lämmle und M. Huber, „Simulation-driven machine learning for robotics and automation," tm - Technisches Messen, pp. 673-684, August 2019.

https://doi.org/10.30844/I40M_21-2_S25-29