Bei diesem Beitrag handelt es sich um einen wissenschaftlich begutachteten und freigegebenen Fachaufsatz („reviewed paper“).

doi.org/10.37544/1436-4980-2025-03-67

Enabler für wandlungsfähige Robotersysteme im öffentlichen Raum und in der Industrie

Onlinebahnplanung mittels Reinforcement Learning

J. Möhrle, A. Gaugenrieder, C. Härdtlein, R. Daub

ZUSAMMENFASSUNG In Produktionen mit hoher Variantenvielfalt und kurzen Produktlebenszyklen steigert Onlinebahnplanung die Wandlungsfähigkeit autonomer Roboter. Mittels Reinforcement Learning (RL) sind Roboter in der Lage, sich dynamisch an variierende Bedingungen anzupassen und komplexe Tätigkeiten zu automatisieren. Dieser Beitrag erläutert die Grundlagen von Onlinebahnplanung mittels RL, stellt ein Konzept anhand der Abfallsammlung im öffentlichen Raum vor und diskutiert dessen Transfer in die Industrie.

STICHWÖRTER

Automatisierung, Künstliche Intelligenz, Flexible Fertigungssysteme

Online path planning using Reinforcement Learning - Enabler for versatile robot systems in public spaces and industry

ABSTRACT Flexible path planning for autonomous robots is required in productions with a high degree of variability and short product life cycles. Reinforcement Learning (RL) offers a solution, as it enables robots to adapt dynamically to varying conditions and automate complex activities. The article explains the basics of online path planning using RL, presents a concept based on waste collection in public spaces, and discusses its transfer to industry.

1 Motivation

Aktuell ist der industrielle Sektor mit zwei Herausforderungen konfrontiert. Zum einen erfordern die sozioökonomischen Entwicklungen des Fachkräftemangels und des demografischen Wandels eine weitgehende Automatisierung von Prozessabläufen. Zum anderen steigen die Anforderungen an produzierende Unternehmen in Form von kürzer werdenden Produktlebenszyklen und steigender Produktvarianz. Als Resultat benötigt es autonome Systeme, die zur variantenreichen Montage befähigt sind. [1]

Die aufwandsarme Anpassung von Betriebsmitteln an sich verändernde Produkte, Prozesse oder Mengen wird in der Literatur als Wandlungsfähigkeit bezeichnet. Ein Wandlungsbefähiger ist etwa die Universalität von Betriebsmitteln [2]. Industrieroboter werden als gesteuerte, frei programmierbare Universaloperatoren in mindestens drei Achsen eingesetzt, um heterogene Handhabungsaufgaben zu automatisieren [3]. Jedoch ist in der industriellen Praxis der Betrieb von Industrierobotern an manuelle Tätigkeiten gekoppelt. Der Arbeitsablauf des Robotersystems wird meist als Teil des Rüstprozesses durch Handführung (Playback) oder Teach-in mittels Programmierhandgerät festgelegt [4].

Die manuelle Anpassung des Robotersystems an sich verändernde Umgebungsbedingungen ist ein Kostentreiber, da die Anpassungsaufwände zu Verfügbarkeitsverlusten der Produktionsanlage führen und Fachkräfte binden. Dabei ist der Roboter jeweils auf die starr programmierten Rahmenbedingungen limitiert. Daraus ergibt sich der Bedarf nach einem wandlungsfähigen Robotersystem, das im Betrieb (online) selbstständig die Roboterbahn unter Beachtung von sich stetig verändernden Umgebungsbedingungen generiert. Mit dieser Aufgabenstellung beschäftigt sich die Bahnplanung in der Robotik. Reinforcement Learning (deutsch: Bestärkendes Lernen), ein auf künstlicher Intelligenz basierender Regelungsansatz, bietet hierfür Potenzial.

Im Folgenden wird zunächst der Stand der Technik im Bereich der Bahnplanung dargelegt. Daraus wird das Potenzial von Reinforcement Learning in der Bahnplanung abgeleitet und ein grundlegendes Verständnis dieses Paradigmas geschaffen. Im Anschluss wird ein Forschungsansatz für einen Reinforcement Learning-Agenten vorgestellt und zuletzt der Transfer in die Industrie diskutiert.

2 Stand der Technik

Bahnplanung bezeichnet in der Robotik die Problembeschreibung zur Findung eines Zielzustands s_{Ende}, ausgehend von einem Startzustand s_{Start} in einem hindernisfreien Raum S_{Frei} unter Einhaltung von Beschränkungen, wie Gelenklimits oder Momentengrenzen [5]. Die Generierung einer solchen Bahn ist als nicht lösbar in deterministischer Polynomialzeit (englisch: Nondeterministic Polynomial Time, kurz: NP-Schwer) zu klassifizieren, da kein bekannter Algorithmus einen optimalen Pfad in effizienter Zeit extrahieren kann [6].

Es wird differenziert zwischen der Offlinebahnplanung, die in einem vorgelagerten Schritt unter Kenntnis der gesamten Umgebung abläuft, und der Onlinebahnplanung, die während des Betriebs abläuft. Im Folgenden soll explizit das Potenzial diverser Bahnplanungsalgorithmen für die Onlinebahnplanung adressiert werden, da das Robotersystem nur mittels Onlinebahnplanung

ohne Verfügbarkeitsverluste auf sich verändernde Umgebungen reagieren kann [7, 8].

2.1 Bahnplanungsalgorithmen

Stichprobenbasierte Algorithmen erreichen ein zeiteffizientes Verhalten, indem die Umgebung stichprobenartig abgetastet wird [6].

Probabilistic Roadmaps (PRM) [9] erstellt zu Beginn stichprobenartig N Abtastpunkte im Arbeitsraum S des Robotersystems. Die k-nächsten Abtastpunkte werden durch Kanten verbunden, sofern sie sich im hindernisfreien Raum S_{Frei} befinden [5, 6]. Dieses Prinzip ist für einen zweidimensionalen Anwendungsfall in **Bild 1** a) dargestellt. Durch eine nachfolgende Graphensuche, beispielsweise A-Star [10], kann ein optimaler, kollisionsfreier Pfad aus der Roadmap extrahiert werden [11]. Dieser ist in Bild 1 a) als unterbrochene Linie visualisiert.

Rapidly Exploring Random Trees (RRT) [12] generiert schrittweise eine Baumstruktur, beginnend beim Startzustand s_{Start} zum Endzustand s_{Ende}. Dabei wird jeweils der nächstgelegene Punkt der Baumstruktur zum Abtastpunkt durch eine Kante verknüpft. In einer nachfolgenden Kollisionsprüfung werden Kanten, die ein Hindernis berühren, entfernt [6]. Gegenüber PRM werden bei RRT die Abtastpunkte nicht einmalig, sondern iterativ generiert. Dies erlaubt eine effizientere Abtastung des Arbeitsraums durch die Nutzung von bereits gewonnenem Wissen, beispielsweise mittels Heuristiken, wie in Bild 1 b) skizziert. [13]

Demgegenüber verfolgt der Algorithmus des künstlichen Potenzialfeldes (englisch: Artificial Potential Field, APF) [14] einen echtzeitfähigen Ansatz, der auf einer sensorischen Erfassung der Abstände umliegender Kollisionsgeometrien relativ zum Roboter basiert. Auftretende Hindernisse induzieren auf den Roboter eine „Abstoßungskraft“ und das Ziel eine „Anziehungskraft“, wie in Bild 1 c) durch Kraftvektoren veranschaulicht [6, 14]. Durch diese imaginären Kraftanteile wird der Arbeitsraum des Roboters mit einem künstlichen Potenzialfeld gefüllt. Das Potenzial kann dabei als Summierung aller auf den Roboter wirkenden Kräfte bei der Bewegung des Roboters durch das

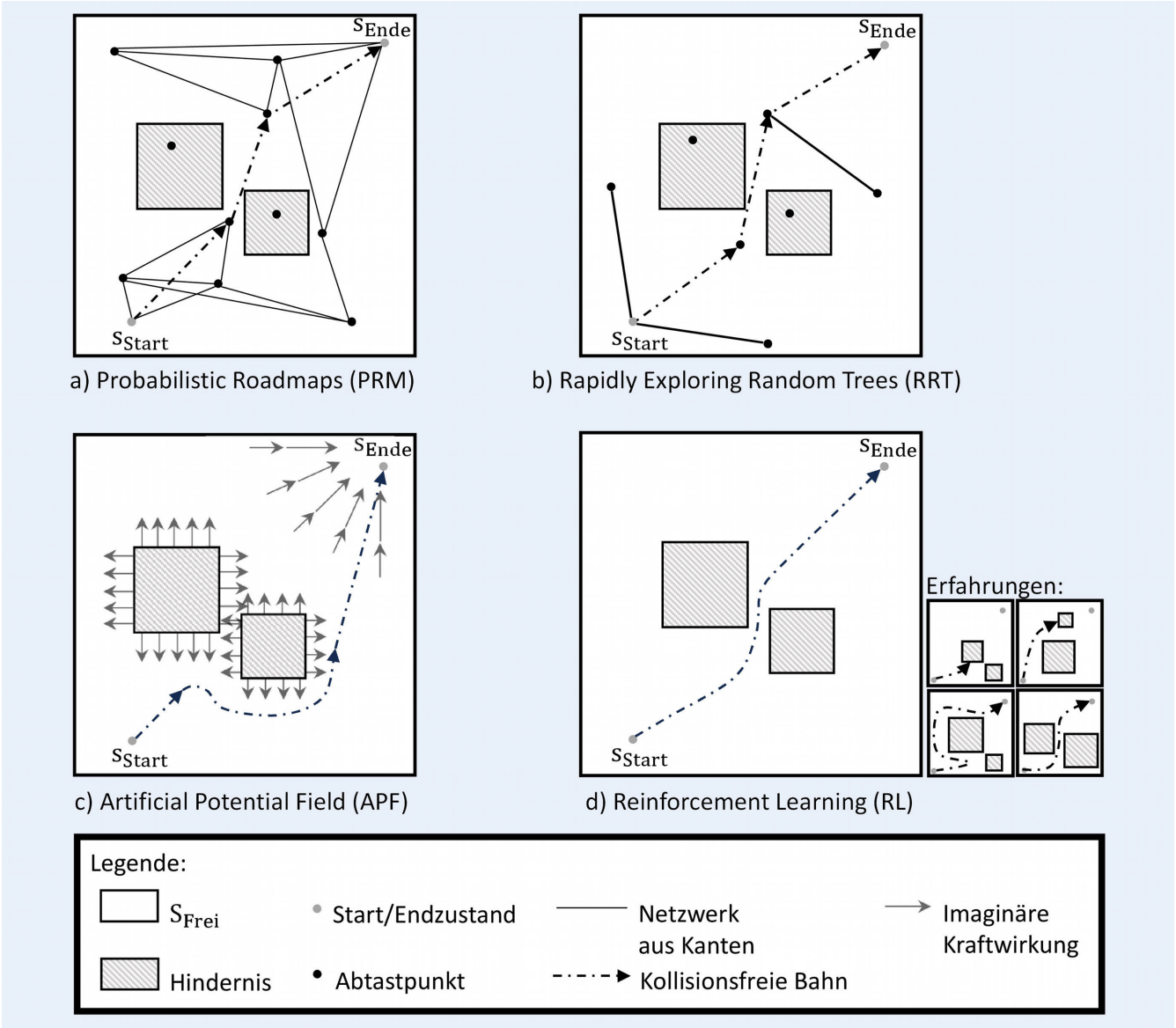

Bild 1. Visualisierung der Funktionsweise bestehender Bahnplanungsalgorithmen. *Grafik: Fraunhofer IGCV*

Potenzialfeld interpretiert werden. Die Optimierung dieses Potenzials ergibt die gewünschte Bahn. Eine diskrete Abtastung der unbekannten Umgebung ist aufgrund der kontinuierlichen sensorischen Rückkopplung nicht nötig. Allerdings tendiert die APF-Methode zu einer Konvergenz in lokalen Minima, verursacht durch lokale Kräftegleichgewichte der imaginären Kraftvektoren. [15]

Das bestärkende Lernen (Reinforcement Learning, RL) ist ein Ansatz der künstlichen Intelligenz. Durch Interaktion mit der Umgebung entwickelt der RL-Agent eine Bewegungsstrategie, um kollisionsfrei vom Startzustand s_{Start} zum Zielzustand s_{Ende} zu gelangen [16]. Analog zum APF-Ansatz kann auf Basis von Sensorwerten echtzeitfähig auf variierende Umgebungsbedingungen reagiert werden [17]. So belegen Untersuchungen von *Raajan et al.* [18], dass RL gegenüber traditionellen stichprobenbasierten Verfahren die Rechenzeit um den Faktor 120 reduziert. Die Bewegungsstrategie wird nicht ausschließlich aus Sensorwerten abgeleitet, sondern durch einen Lernprozess basierend auf den historischen Interaktionen mit der Umgebung angepasst. Dies ist in Bild 1 d) durch gespeicherte Erfahrungen visualisiert. Dadurch ist die künstliche Intelligenz, ähnlich dem menschlichen Verhalten, dazu befähigt, aus vergangenen Misserfolgen oder Erfolgen durch Versuch und Irrtum zu lernen. RL-Bahnplanung hebt sich somit vom APF-Ansatz ab: Aus den gesammelten Erfahrungswerten können komplexe Szenarien erfolgreich bewältigt werden.

Ein Überblick über Bahnplanungsmethoden des Stands der Technik verdeutlicht das Alleinstellungsmerkmal von RL. Stichprobenbasierte Algorithmen wie PRM oder RRT benötigen bei sich ändernder Umgebung eine Neuerstellung oder Aktualisierung der diskretisierten Repräsentation des Arbeitsraums. Die Anzahl an Abtastpunkten limitiert die Feinheit der Bahn, bedingt jedoch höhere Rechenaufwände [19]. Stichprobenbasierte Algorithmen sind daher für reaktive Kollisionsvermeidung ungeeignet [15]. Der APF-Ansatz ist durch die sensorbasierte Rückkopplung echtzeitfähig, kann jedoch lokale Minima im Potenzialfeld nicht beherrschen. RL lernt aus der historischen Interaktion mit dem Umfeld und kann daraus intelligente Strategien ableiten, um beliebige Kollisionsszenarien zu überwinden.

2.2 Reinforcement Learning in der Bahnplanung

Das RL-Paradigma beschreibt einen Agenten mit der Aktion a_t als Output sowie den beiden Inputs: Zustand s_t und Belohnung r_t (englisch: reward), jeweils zum Zeitpunkt t [16]. Durch die Anordnung von a, s und r in einem Regelkreis wird ein reaktives System geschaffen, zu sehen in **Bild 2** a).

Die Aktion a_t des Agenten manipuliert die Umgebung und induziert eine Zustandsänderung $s_t \rightarrow s_{t+1}$. Die Relation des neuen Zustands s_{t+1} zu einem Zielzustand s_{Ende} wird durch die Beloh-

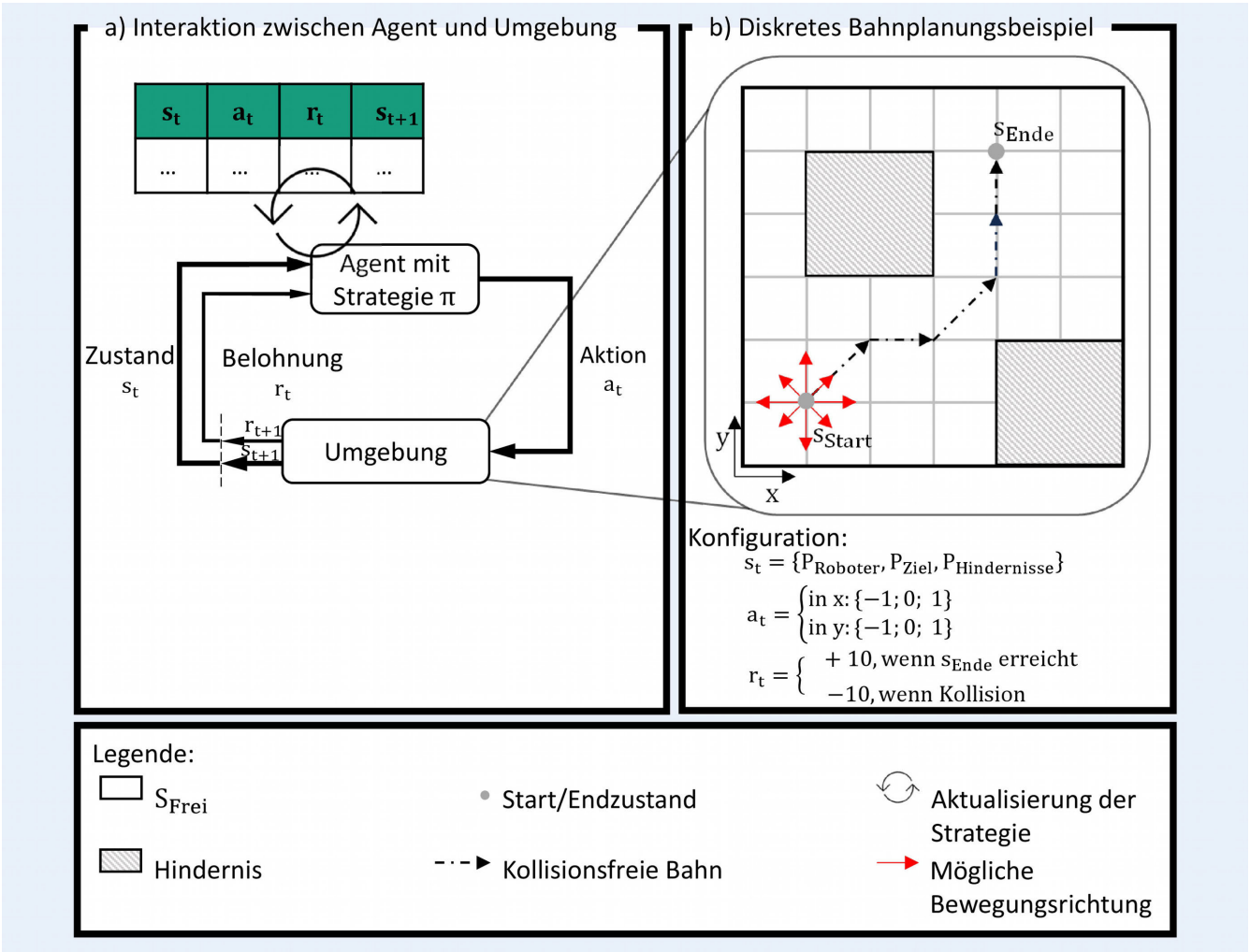

Bild 2. Anwendung von Reinforcement Learning in der Bahnplanung. *Grafik: Fraunhofer IGCV*

nung r_t quantifiziert. Gemäß einer Regelungsstrategie π (englisch: policy) des Agenten resultiert im nachfolgenden Zeitschritt die Aktion a_{t+1} basierend auf der aktuellen Zustandsbeschreibung s_{t+1}. Der RL-Agent optimiert kontinuierlich die „intelligente" Regelungsstrategie π, um die kumulative Belohnung $r_{t\rightarrow ti}$ einer festgelegten Zeitspanne t_i zu maximieren. [16] Dieser Optimierungsprozess bildet den Trainingsprozess der künstlichen Intelligenz. Basis dieser Überlegungen ist der Markov-Entscheidungsprozess, welcher besagt, dass eine Zustandsänderung $s_t \rightarrow s_{t+1}$ eines Systems vollständig durch den aktuellen Zustand s_t und der Aktion a_t beschrieben wird. Diese Voraussetzung ist in realen Applikationen nicht zwangsläufig gegeben, da oft nicht alle Informationen für eine vollständige Beschreibung des Zustands physikalisch erfasst werden können. [20]

Die Anwendung des RL-Paradigmas in der Bahnplanung ist in Bild 2 b) beispielhaft für einen zweidimensionalen Anwendungsfall skizziert. Ziel ist die Findung eines kollisionsfreien Pfades von s_{Start} zu s_{Ende} innerhalb S_{Frei}. In jedem Zeitschritt entscheidet der Agent über die Bewegung des Massenschwerpunktes durch eine diskrete Aktion $a = \{-1;0;1\}$ sowohl in x- als auch in y-Richtung – der Agent kann sich senkrecht, horizontal oder diagonal bewegen. Der aktuelle Zustand wird durch Messgrößen wie etwa Position des Massenschwerpunktes, des Ziels und der Hindernisse beschrieben. Die Belohnungsfunktion quantifiziert die Erfüllung des Zielzustands in jedem Zeitschritt durch einen positiven Anteil bei Zielerreichung und einen negativen Anteil bei Kollisionen mit Hindernissen.

In jedem Zeitschritt wird das Tupel $\{s_t, a_t, r_t, s_{t+1}\}$ gespeichert. Diese Informationssammlung der Vergangenheit erlaubt eine Optimierung der Bahnplanungsstrategie π über den Trainingsprozess hinweg. In der Robotik wird die Strategie π meist ohne Kenntnis eines Modells der Umgebung, sondern durch ein neuronales Netz generiert, sogenannte modellfreie Ansätze. [16, 21]

Beim Training des RL-Agenten besteht ein struktureller Konflikt, das Exploration-Exploitation-Dilemma: Der Agent kann seine bestehende Strategie π nur auf Basis vergangener Interaktionen mit der Umgebung optimieren: das Ausnutzen (englisch: exploitation) von Wissen. Um diese historischen Interaktionen zu erhalten, müssen zunächst eine Vielzahl von Erfolgen und Misserfolgen in der Umgebung gesammelt werden: das Erkunden (englisch: exploration) der Umgebung. Beispielsweise wird der RL-Agent nur durch vermehrte Kollisionen in Bild 2 b) deren negative Wirkung abbilden können. [16]

Eine Balance von Erkundung der Umgebung und Ausnutzung der vorhandenen Wissensbasis ist für einen effizienten Trainingsprozess erforderlich. Abhängig von der Aufgabenstellung und der Komplexität der Umgebung werden folglich mehrere hunderttausend Zeitschritte zum Training benötigt [16]. Aus Gründen der Zeiteffizienz wird ein RL-Agent in Simulationen trainiert [21]. Abschließend muss der in Simulationen trainierte RL-Agent in die reale Applikation transferiert werden [22]. Aufgrund von Abweichungen zwischen Simulation und Realität (Sim2Real gap) ist ein nahtloser Transfer nicht möglich [23].

Die Auslegung des RL-Agenten ist nicht trivial und benötigt Expertenwissen:

- Der Arbeitsraum ist in Anbetracht der vorhandenen Aktorik zu definieren, wobei zwischen kontinuierlich und diskret unterschieden wird.
- Der Zustand des Systems sollte durch reale Messgrößen vollständig sowie eindeutig beschrieben werden. So ist in Bild 2 b) die Position der Hindernisse $P_{Hindernisse}$ nur bei konstanten Kollisionsgeometrien aussagekräftig, da nur die Position des Flächenschwerpunkts erfasst wird.
- Die Belohnungsfunktion sollte bestenfalls durch eine kontinuierliche Funktion den Grad der Zielerfüllung eindeutig beschreiben [24]. Unpräzise Formulierungen führen zu einer ungewollten Verhaltensweise des Agenten, beispielsweise wird die Belohnungsfunktion in Bild 2 b) durch ständiges Annähern und Entfernen relativ zum Ziel maximiert, anstatt einer möglichst effizienten Zielerreichung in kurzer Zeit.
- Hyperparameter sind spezifisch nach Algorithmus und Anwendungsfall zu selektieren, welche unter anderem die Konvergenz des Trainings (Lernrate), die Wichtung zukünftiger Belohnungen (Discount-Faktor) oder die Aggressivität der Umgebungsexploration (Explorationsrate) beeinflussen. [16]

Aufgrund dieser Komplexität zur Generierung eines RL-Agenten ist Onlinebahnplanung mittels RL nur für hochflexible Umgebungen geeignet. Bei einfachen, starren Bahnplanungsproblemen ist derzeit auf klassische Bahnplanungsalgorithmen, wie PRM oder RRT, zurückzugreifen. [17, 25]

Diverse Institutionen erforschen Onlinebahnplanung mittels RL. In der mobilen Robotik wird die Interaktion mit unbekannten Umgebungen, wie etwa in der Logistik oder für die Off-Road-Navigation, betrachtet [26]. Mobile Roboter werden dabei ähnlich zu Bild 2 b) als Massenschwerpunkt geregelt. Demgegenüber müssen bei Gelenkarmrobotern, wie sie in klassischen Roboterzellen zu finden sind, die kinematischen Beziehungen zwischen den Gliedern berücksichtigt werden [5]. Dies resultiert in einer Regelungsaufgabe im dreidimensionalen Raum mit gekoppelten Aktoren. Durch die erweiterte Dimension des Aktionsraums wächst die Komplexität des Bahnplanungsproblems gegenüber der mobilen Robotik exponentiell an [6].

Die Forschung adressiert hier Ansätze zum einfachen Greifen von Objekten ohne Betrachtung von Hindernissen [27], zur Interaktion mit statischen Hindernissen [28] oder der sicheren Interaktion mit Menschen (Mensch-Roboter-Kooperation) [21]. Die Komplexität des RL-Agenten wird zum Beispiel durch manuelle Expertendemonstrationen in der Simulation [29] oder einer regelbasierten Selektion zwischen einer RL-Onlinebahnplanung und einer stichprobenbasierten Offlinebahnplanung [30] reduziert. Der Stand der Technik beschränkt sich auf simulative Erprobungen oder Validierungen in Labormaßstäben [31]. Anwendungsorientierte Ansätze, die alle umliegenden Systemeigenschaften eines realen Anwendungsfalls, wie Sensorik oder individuelle Roboterkinematiken, betrachten, sind nicht vorhanden. Zudem wird das Potenzial, die Komplexität des RL-Agenten durch Systemwissen zu reduzieren, bisher nicht ausgeschöpft.

3 Reinforcement Learning-Agent zur autonomen Abfallsammlung

Onlinebahnplanung mittels RL ist ein aktueller Forschungsgegenstand des Fraunhofer Institut für Gießerei-, Composite-, und Verarbeitungstechnik IGCV im Rahmen des Forschungsvorhabens AutASa „Automatisiertes Abfallsammelfahrzeug".

Das Vorhaben befasst sich in der Gesamtheit mit der Entwicklung eines Prototyps zum autonomen Handling von Mülltonnen in beliebiger Anordnung. Aufgrund der dynamischen Umgebungseigenschaften mit ständig variierenden Hindernissen, wie Laternen, Bäumen oder Kraftfahrzeugen, sowie sich ändernden

Positionen der Mülltonnen, ist ein RL-Agent prädestiniert zur Befähigung der Autonomie.

3.1 Anforderungen an einen Reinforcement Learning-Agenten zur autonomen Abfallsammlung

Das Handling der Mülltonnen erfolgt durch eine an Abfallsammelfahrzeuge angepasste Leichtbauroboterkinematik, entwickelt durch die Teon GmbH. Mittels Stereokamerasysteme der Roboception GmbH wird die Umgebung des Roboters erfasst. Basierend auf dieser digitalen Repräsentation des Umfelds in Form einer Punktwolke entwickelt das Fraunhofer IGCV eine RL-Onlinebahnplanung zum Abgreifen von Mülltonnen, Zuführen in die Entleerungsmechanik des Abfallsammelfahrzeugs sowie nachfolgendem Abstellen der entleerten Mülltonnen. Die MRK-Systeme GmbH koordiniert das gesamte Vorhaben und detailliert das Sicherheitskonzept des Robotersystems.

Anforderungen an die Bahnplanung sind ein kollisionsfreies Manövrieren durch den öffentlichen Raum unter Beachtung ständig variierender Kollisionsszenarien und Zielposen. Die Bahnplanung muss mit den Systemeigenschaften hinsichtlich Kinematik und Vision-System kompatibel sein. Bestehende RL-Agenten sind für die spezifischen Anforderungen der kommunalen Abfallsammlung und der resultierenden Greifstrategie nicht geeignet (vergleiche Kapitel 2.2). Im Folgenden wird der Workflow zum Training des RL-Agenten sowie die Greifstrategie erläutert.

3.2 Workflow zum Training des Reinforcement Learning-Agenten zur autonomen Abfallsammlung

Aus Gründen der Zeiteffizienz erfolgt das Training des RL-Agenten in einer physikbasierten Simulationsumgebung. Anders als rein visuelle Simulationsumgebungen verfügt diese über eine Physikengine zur Modellierung physikalischer Eigenschaften, wie Massenträgheiten oder Kollisionen [23]. Dennoch bestehen Abweichungen zwischen Simulation und Realität, welche auch als „Sim2Real gap“ bezeichnet werden. Als Ursachen sind etwa geometrische Abweichungen durch Fertigungsungenauigkeiten oder temperaturbedingte Längenänderungen zu nennen. Auch weisen simulierte Sensordaten gegenüber realen Sensordaten eine systematische Abweichung aufgrund von Messrauschen auf. [23]

Diese Diskrepanzen von Simulation und Realität erfordern einen Ansatz zur Reduktion des Sim2Real gap, welcher in **Bild 3** dargestellt ist.

Die abgebildete SCARA-Kinematik (Selective Compliance Assembly Robot Arm) dient zur exemplarischen Veranschaulichung des Sim2Real gap.

1. CAD des Roboters: In den CAD (Computer Aided Design)-Daten werden durch den Konstrukteur die kinematischen sowie dynamischen Eigenschaften des Roboters festgelegt. Im Forschungsprojekt AutASa wurde eine neuartige Fünf-Achs-Kinematik entwickelt (vergleiche Kapitel 3.3 Greifstrategie).
2. URDF des Roboters: Aus den CAD-Konstruktionsdaten wird eine URDF (Unified Robot Description Format) Datei abge-

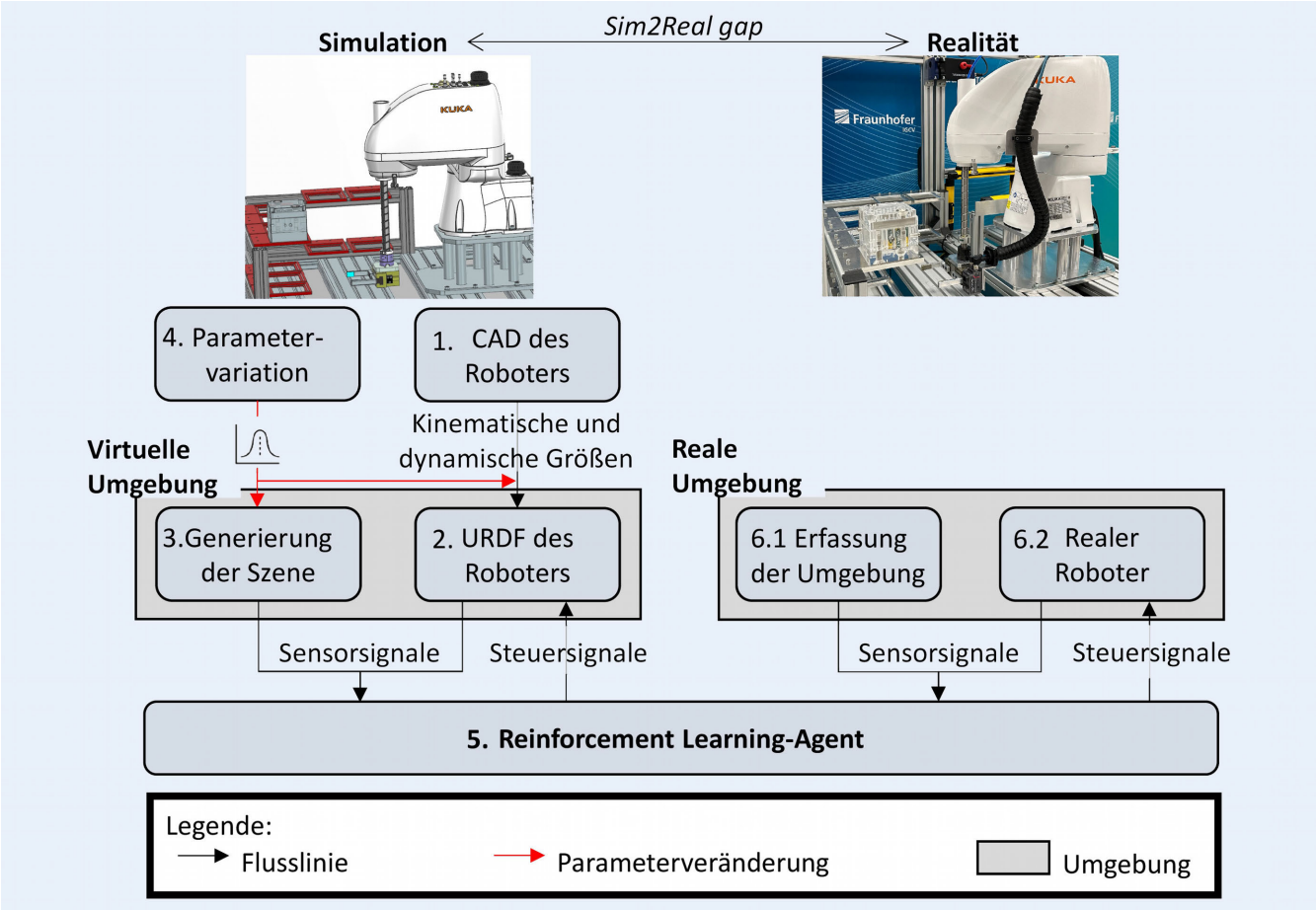

Bild 3. Ansatz zur Übertragung des Reinforcement Learning-Agenten in die Realität. *Grafik: Fraunhofer IGCV*

leitet. Dies ist ein auf XML (Extensible Markup Language) basierendes Dateiformat zur Beschreibung der Kinematik und Dynamik von Robotersystemen in Form einer Baumstruktur. Dabei wird die Verbindung der kinematischen Glieder (Karussell → Schwinge → Arm → Handgelenk) durch Gelenktypen in Eltern-Kind-Beziehungen beschrieben und geometrische sowie dynamische Eigenschaften zugewiesen.

3. Generierung der Szene: Der Roboter wird mittels URDF modelliert und in eine aufgabenspezifische, simulierte Szene inkludiert. Im beschriebenen Anwendungsfall des Forschungsprojekts AutASa besteht diese aus randomisierten Positionen von Mülltonnen sowie Kollisionsgeometrien.
4. Parametervariation: Zur Kompensation der Abweichungen zwischen Simulation und Realität wird eine systematische Variation von Modellierungsparametern vorgenommen (Domain Randomization) [32], um die Robustheit des RL-Agenten zu steigern. Diese Manipulation der Modellierungsparameter ist in Bild 3 rot dargestellt und beeinflusst sowohl die kinematischen und dynamischen Parameter der URDF-Datei als auch die Sensorwerte.
5. Reinforcement Learning-Agent: In der Simulationsumgebung lernt der Agent eine Bewegungsstrategie, um das Zielobjekt durch den Roboter kollisionsfrei zu erreichen. Die Konvergenz des Trainingsprozesses hängt von der gewählten Konfiguration (Aktionsraum, Zustandsraum, Belohnungsfunktion) ab.
6. Reale Umgebung: Nach Vollendung des Trainingsprozesses in der Simulation, wird der RL-Agent auf die reale Umgebung appliziert:
   6.1 Die simulierten Sensorsignale werden durch reale Messgrößen substituiert.
   6.2 Ein reales Robotersystem wird anstatt der URDF-Datei geregelt.

3.3 Greifstrategie

Im Folgenden wird die Greifstrategie anhand des Fallbeispiels der autonomen Abfallsammlung im öffentlichen Raum des Forschungsprojekts AutASa vorgestellt. Die Greifstrategie muss die kollisionsfreie Interaktion mit der Umgebung, unter Einhaltung der kinematischen Struktur sowie der sensorischen Erfassung des Umfelds, erfüllen.

Das Stereokamerasystem erfasst die reale Welt im Bildraum als Pixel. Bei weit entfernten Objekten stehen somit weniger Pixel zur Verfügung, um Details der realen Welt zu repräsentieren. Das Pixel/mm-Verhältnis limitiert die Genauigkeit des Bildverarbeitungssystems. Denn basierend auf der Repräsentation im Bildraum wird die Position des Zielobjekts im Koordinatensystem des Roboters berechnet (Hand-Auge-Kalibrierung). Diese Position ist eine Eingangsmessgröße für den RL-Agenten. Um trotz des Pixel/mm-Verhältnisses das gesamte Arbeitsumfeld mit einem Arbeitsradius von circa 2,5 m ganzheitlich zu erfassen und gleichzeitig die Zielposition mit ausreichender Genauigkeit zu extrahieren, wird eine zweistufige Greifstrategie verfolgt (wie in **Bild 4** dargestellt):

- Bahnplanung Nr. 1: Erreichung der Approach-Position unter Verwendung der festen Stereokamera am Abfallsammelfahrzeug durch RL
- Bahnplanung Nr. 2: Feinpositionierung durch Point-To-Point (PTP) und Linearbewegung (LIN) unter Verwendung der lokalen Stereokamera am Greifer

Diese beiden Bahnplanungsaufgaben sollen im Forschungsprojekt AutASa durch eine Fünf-Achs-Kinematik realisiert werden. Fünf Achsen gewährleisten eine Bewegungsflexibilität im dreidimensionalen Raum, wobei zwei translatorische Achsen die nötige Steifigkeit zum Handling befüllter Mülltonnen mit einem Gewicht bis zu 160 kg bieten. Die kinematische Struktur sowie deren Zuordnung zur Bahnplanung zeigt Bild 4:

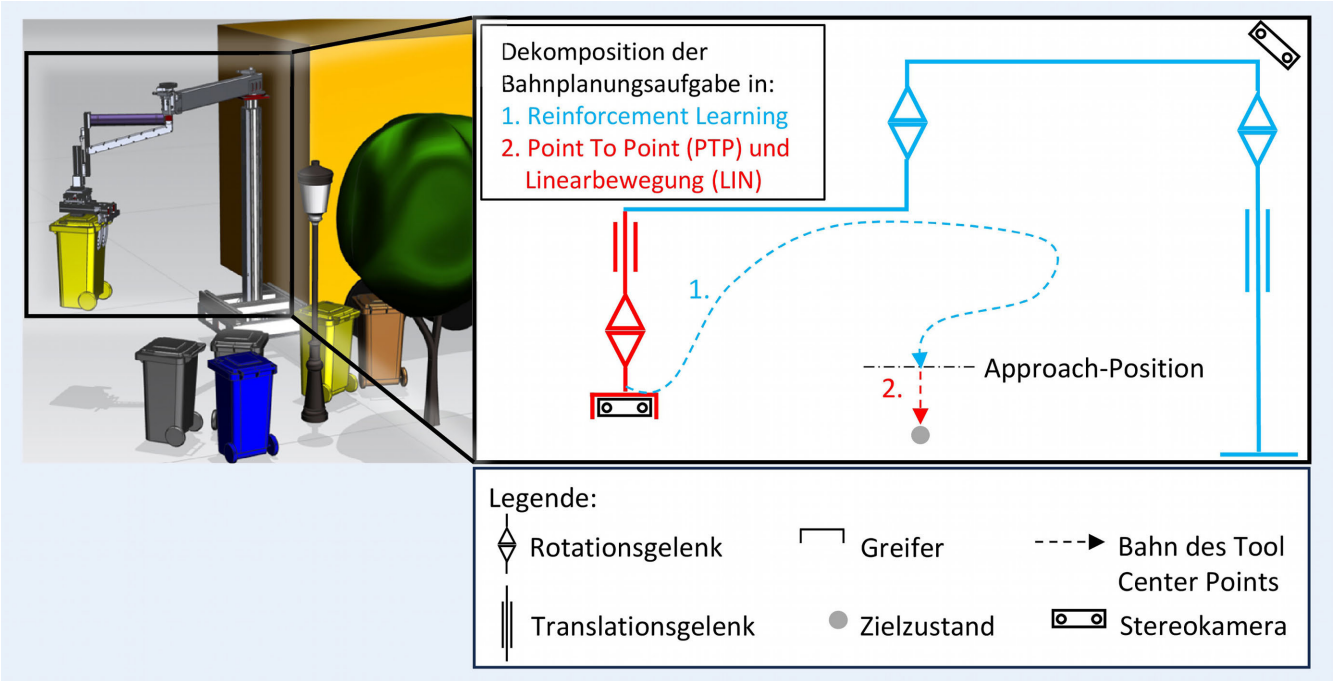

Bild 4. Greifstrategie zur autonomen Abfallsammlung durch Reinforcement Learning mittels neu-konzeptionierter Fünf-Achs-Kinematik.
Grafik: Fraunhofer IGCV

1. Die blau gekennzeichneten Teile der Kinematik, ein translatorischer Großhub und zwei rotatorische Freiheitsgrade, werden mittels des RL-Agenten zur Erreichung der Approach-Position geregelt. Hierdurch können im dreidimensionalen Raum Ausweichmanöver abgebildet werden. Die Umgebungsinformationen werden durch die feste Stereokamera am Abfallsammelfahrzeug erfasst. Die Approach-Position wird dem RL-Agenten als Eingangsgröße übermittelt und ist abhängig vom Greifobjekt dadurch definiert, dass sich das Zielobjekt im Nahsichtfeld der lokalen Stereokamera im Greifer des Roboters befindet.
2. Durch die letzten zwei kinematischen Glieder (Translationsgelenk, Rotationsgelenk), in Bild 4 rot markiert, wird nachfolgend eine Feinpositionierung mittels PTP- und LIN-Befehl auf Basis der Hand-Auge-Kalibrierung mit der lokalen Stereokamera durchgeführt.

Dieser hybride Regelungsansatz reduziert die RL-Bahnplanung auf drei Freiheitsgrade. Die verringerte Anzahl an geregelten Achsen hat eine Reduktion des Aktionsraums sowie Zustandsraums zur Folge, was die Datenintensivität des Trainingsprozesses schmälert. Ebenfalls vereinfacht die Feinpositionierung mittels konventioneller Roboterbefehle den Transfer des RL-Agenten von Simulation auf den realen Roboter, da marginale Abweichungen zwischen simulierter Welt und realer Welt zur Erreichung der Approach-Position unkritisch sind.

Der präzise Greifprozess wird durch einen kamerabasierten PTP/LIN-Befehl, vergleichbar mit einem Griff aus der Kiste (englisch: Bin Picking), umgesetzt. Der hochflexible, aber trainingsintensive Bahnplanungsansatz via RL wird somit auf dessen Wirksamkeit zur Interaktion in heterogenen Umgebungen beschränkt. Das Forschungsvorhaben verfolgt das anwendungsorientierte Credo „so einfach wie möglich, so kompliziert wie nötig“. In diesem Kontext ist die Greifstrategie als ein inhärenter Baustein des Sim2Real-Transfers gemäß Bild 3 zu betrachten.

4 Transfer in die Produktionstechnik

Wie eingangs motiviert, resultiert die steigende Variantenvielfalt im industriellen Sektor in einer volatilen Produktionsumgebung. Besonders bei kleiner werdenden Losgrößen disqualifizieren sich klassische Bahnplanungsalgorithmen (RRT, PRM) aufgrund von Verfügbarkeitsverlusten. Für diese hochflexiblen Anforderungen ist Onlinebahnplanung mittels RL für industrielle Anwendungen zukünftig zu betrachten [25]. Der in diesem Beitrag vorgestellte hybride Ansatz zur Onlinebahnplanung mittels RL ist generalisierbar und kann somit in industrielle Applikationen transferiert werden.

Als Anwendungsfälle in der Industrie für RL-Bahnplanung sind zu nennen:

1. Nutzung eines Robotersystems an heterogenen Anlagen, wodurch flexibel auf Störkonturen reagiert werden muss.
2. Stark variierende Kollisionsgeometrien durch Rüstelemente an homogenen Anlagen.
3. Stark variierende Dimension und Form der Greifobjekte, die eine Anpassung der Bahn zur Kollisionsvermeidung erfordern.
4. Kooperierende Arbeitsformen zwischen Menschen und Roboter.

Entsprechende Anwendungsfälle werden am Fraunhofer IGCV mit dem Demonstrator für wandlungsfähige Produktion abgebildet. Dieser befähigt durch elektromechanische Schnellkupplungen dazu, beliebige Produktionsmodule per Plug-&-Produce an einem linearen Transportsystem zu adaptieren. Dadurch können volatile Produktionsbedingungen, wie Kapazitätsengpässe oder hohe Produktvielfalt, mit minimalen Rüstzeiten bewältigt werden. Robotersysteme werden dabei bedarfsgesteuert heterogenen Produktionsbedingungen mit variierenden Störkonturen zugefügt (Anwendungsfall 1). Je nach Greifobjekt auf dem linearen Transportsystem muss die Bahnplanung online angepasst werden (Anwendungsfall 3). Der Aufbau mit kollaborierenden Robotern erlaubt langfristig eine Erweiterung hin zur Mensch-Roboter-Kooperation (Anwendungsfall 4). Der vorgestellte Bahnplanungsansatz soll zukünftig am Demonstrator implementiert und verfeinert werden.

Um einen Transfer des vorgestellten hybriden Regelungsansatz aus Bild 4, bestehend aus RL und konventionellen Roboterbefehlen (PTP/LIN), auf andere Kinematiken zu ermöglichen, müssen die einzelnen Gelenke aufgabenspezifisch den beiden Anteilen (RL, konventionelle Roboterbefehle) zugeordnet werden. Aufgrund der hohen kinematischen Ähnlichkeit zur Fünf-Achs-Kinematik aus Bild 4 erfolgt die Implementierung am Demonstrator zunächst anhand einer SCARA-Kinematik, dargestellt in Bild 3. Gegenüber der Eigenkonstruktion zur autonomen Abfallsammlung in Bild 4 entfällt das erste translatorische Gelenk. Die Onlinebahnplanung mittels RL kann analog zum Anwendungsfall der autonomen Abfallsammlung durch einen dreidimensionalen Aktionsraum, zusammengesetzt aus einem Translationsgelenk und zwei Rotationsgelenken, erfolgen.

Bei Knickarm-Kinematiken ist aufgabenspezifisch die Approach-Position und basierend auf deren Zugänglichkeit die Zuordnung zu den beiden Anteilen festzulegen. Prädestiniert sind die vorderen Glieder der kinematischen Kette zur Grobpositionierung (Karussell → Schwinge) durch RL und die hinteren Glieder der kinematischen Kette (Arm → Handgelenk) zur Feinpositionierung durch konventionelle Roboterbefehle.

5 Zusammenfassung und Ausblick

Reinforcement Learning (RL) beschreibt ein Paradigma, bei dem ein Agent aus Versuch und Irrtum schrittweise Strategien zur Interaktion mit einer Umgebung ableitet. Dieses Prinzip kann für die Onlinebahnplanung in dynamischen Produktionsumgebungen verwendet werden. Gegenüber klassischen Bahnplanungsmethoden zeichnet sich RL durch Flexibilität und Reaktionsgeschwindigkeit aus.

Allerdings ist vorgelagert zur Anwendung des RL-Agenten ein aufwendiger Trainingsprozess mit mehreren tausend Simulationsdurchläufen nötig. Der Trainingsprozess bedarf Expertenwissen für die Auswahl des RL-Algorithmus inklusive der Hyperparameter sowie der Definition des Zustandsraums, Aktionsraums und der Belohnungsfunktion. Die Generalisierbarkeit des Agenten ist limitiert auf die in der Simulation präsentierten Szenarien. Ebenfalls erschweren Abweichungen zwischen Realität und Simulation (Sim2Real gap) einen nahtlosen Übergang in industrielle Applikationen.

Aufgrund dieser Aufwände ist Onlinebahnplanung mittels RL derzeit nur in hochflexiblen Produktionsszenarien mit dynamischen Einflussfaktoren, wie dem Menschen, in der Praxis zu empfehlen. Reale Anwendungen für RL-Onlinebahnplanung über Labormaßstäbe hinaus sind noch nicht bekannt.

Dieser Beitrag zeigt einen anwendungsorientierten Ansatz zur Onlinebahnplanung mittels RL. Er wird vorgestellt anhand eines

Anwendungsfalls der kommunalen Abfallsammlung. Neben einem Workflow zur Reduktion des Sim2Real gap, wurde ebenfalls eine neuartige Greifstrategie vorgestellt. Die Bahnplanungsaufgabe wird dabei durch Expertenwissen in Anteile für RL und konventionelle Roboterbefehle zerlegt. Die Anteile werden kinematischen Gliedern des Roboters zugeordnet, wodurch sich die Komplexität des RL-Agenten wesentlich reduziert. Die Ergebnisse adressieren primär Robotik im öffentlichen Raum, bieten jedoch Potenzial zum Transfer in den industriellen Sektor.

FÖRDERHINWEIS

Das Projekt „AutASa“ wird durch das Bundesministerium für Bildung und Forschung gefördert.

Literatur

[1] Reinhart, G.: Handbuch Industrie 4.0. Geschäftsmodelle, Prozesse, Technik. München: Carl Hanser Verlag 2017
[2] Wiendahl, H.-P.; Wiendahl, H.-H.: Betriebsorganisation für Ingenieure. München: Carl Hanser Verlag 2020
[3] DIN EN ISO 10218–1: Robotik – Sicherheitsanforderungen – Teil 1: Industrieroboter (ISO/DIS 10218–1.2:2021), Ausgabe September 2021
[4] Hesse, S.; Malisa, V. (Hrsg.): Taschenbuch Robotik, Montage, Handhabung. München: Carl Hanser Verlag 2016
[5] Lynch, K. M.; Park, F. C.: Modern robotics – mechanics, planning, and control. Cambridge: Cambridge University Press 2017
[6] Yang, L.; Qi, J.; Song, D. et al.: Survey of Robot 3D Path Planning Algorithms. Journal of Control Science and Engineering (2016), pp. 1–22, doi.org/10.1155/2016/7426913
[7] Liu, L.; Wang, X.; Yang, X. et al.: Path planning techniques for mobile robots: Review and prospect. Expert Systems with Applications 227 (2023), #120254
[8] Kroger, T.; Wahl, F. M.: Online Trajectory Generation: Basic Concepts for Instantaneous Reactions to Unforeseen Events. IEEE Transactions on Robotics 26 (2010) 1, pp. 94–111
[9] Kavraki, L. E.; Svestka, P.; Latombe, J.-C. et al.: Probabilistic roadmaps for path planning in high-dimensional configuration spaces. IEEE Transactions on Robotics and Automation 12 (1996) 4, pp. 566–580
[10] Hart, P.; Nilsson, N.; Raphael, B.: A Formal Basis for the Heuristic Determination of Minimum Cost Paths. IEEE Transactions on Systems Science and Cybernetics 4 (1968) 2, pp. 100–107
[11] Madridano, Á.; Al-Kaff, A.; Martín, D. et al.: 3D Trajectory Planning Method for UAVs Swarm in Building Emergencies. Sensors (Basel, Switzerland) 20 (2020) 3, #642, doi.org/10.3390/s20030642
[12] Steven M. LaValle: Rapidly-Exploring Random Trees: A New Tool for Path Planning. Stand:1998. Internet: lavalle.pl/papers/Lav98c.pdf. Zugriff am 04.03.2025
[13] Shahabi, M.; Ghariblu, H.; Beschi, M.: Comparison of different sample-based motion planning methods in redundant robotic manipulators. Robotica 40 (2022) 9, pp. 3104–3119
[14] Khatib, O.: Real-time obstacle avoidance for manipulators and mobile robots. 1985 IEEE International Conference on Robotics and Automation, St. Louis, MO, USA, 1985, pp. 500–505
[15] Liu, H.; Qu, D.; Xu, F. et al.: Real-Time and Efficient Collision Avoidance Planning Approach for Safe Human-Robot Interaction. Journal of Intelligent & Robotic Systems 105 (2022) 4, #93, doi.org/10.1007/s10846–022–01687–0
[16] Sutton, R. S.; Barto, A. G.: Reinforcement learning. An introduction. Cambridge/Massachusetts: The MIT Press 2018
[17] Tan, C. S.; Mohd-Mokhtar, R.; Arshad, M. R.: A Comprehensive Review of Coverage Path Planning in Robotics Using Classical and Heuristic Algorithms. IEEE Access 9 (2021), pp. 119310–119342
[18] Raajan, J.; Srihari, P. V.; Satya, J. P. et al.: Real Time Path Planning of Robot using Deep Reinforcement Learning. IFAC-PapersOnLine 53 (2020) 2, pp. 15602–15607
[19] Chandler, B.; Goodrich, M. A.: Online RRT* and online FMT*: Rapid replanning with dynamic cost. 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Vancouver, BC, Canada, 2017, pp. 6313–6318
[20] Kurniawati, H.: Partially Observable Markov Decision Processes and Robotics. Annual Review of Control, Robotics, and Autonomous Systems 5 (2022) 1, pp. 253–277
[21] Thumm, J.; Althoff, M.: Provably Safe Deep Reinforcement Learning for Robotic Manipulation in Human Environments. arXiv (2022), https://arxiv:2205.06311
[22] Vrabič, R.; Škulj, G.; Malus, A. et al.: An architecture for sim-to-real and real-to-sim experimentation in robotic systems. Procedia CIRP 104 (2021), pp. 336–341
[23] Collins, J.; Howard, D.; Leitner, J.: Quantifying the Reality Gap in Robotic Manipulation Tasks. 2019 International Conference on Robotics and Automation (ICRA), Montreal, QC, Canada, 2019, pp. 6706–6712
[24] Skalse, J.; Howe, N. H. R.; Krasheninnikov, D. et al.: Defining and Characterizing Reward Hacking. arXiv (2022), https://arxiv.org/abs/2209.13085
[25] Cho, J.; Jung, S.: Reinforcement Learning-Based Motion Planning for Robotic Manipulators in Smart Industry. 2024 15th International Conference on Information and Communication Technology Convergence (ICTC), Jeju Island, Republic of Korea, 2024, pp. 1166–1168
[26] Prasuna, R. G.; Potturu, S. R.: Deep reinforcement learning in mobile robotics – a concise review. Multimedia Tools and Applications 83 (2024) 28, pp. 70815–70836
[27] Quillen, D.; Jang, E.; Nachum, O. et al.: Deep Reinforcement Learning for Vision-Based Robotic Grasping: A Simulated Comparative Evaluation of Off-Policy Methods. 2018 IEEE International Conference on Robotics and Automation (ICRA), Brisbane, QLD, 2018, pp. 6284–6291
[28] Xie, J.; Shao, Z.; Li, Y. et al.: Deep Reinforcement Learning With Optimized Reward Functions for Robotic Trajectory Planning. IEEE Access 7 (2019), pp. 105669–105679
[29] Liu, H.; Ying, F.; Jiang, R. et al.: Obstacle-Avoidable Robotic Motion Planning Framework Based on Deep Reinforcement Learning. IEEE/ASME Transactions on Mechatronics 29 (2024) 6, pp. 4377–4388
[30] Sangiovanni, B.; Incremona, G. P.; Piastra, M. et al.: Self-Configuring Robot Path Planning With Obstacle Avoidance via Deep Reinforcement Learning. IEEE Control Systems Letters 5 (2021) 2, pp. 397–402
[31] Tang, C.; Abbatematteo, B.; Hu, J. et al.: Deep Reinforcement Learning for Robotics: A Survey of Real-World Successes. arXiv (2024), https://arxiv.org/abs/2408.03539
[32] Tobin, J.; Fong, R.; Ray, A. et al.: Domain Randomization for Transferring Deep Neural Networks from Simulation to the Real World. 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Vancouver, BC, Canada, 2017, pp. 23–30, doi.org/10.1109/IROS.2017.8202133

Jannik Möhrle, M.Sc.
jannik.moehrle@igcv.fraunhofer.de
Tel. +49 821 / 90678-326
Foto: Fraunhofer IGCV

Andreas Gaugenrieder, M.Eng.

Christian Härdtlein, M.Eng.

Prof. Dr.-Ing. Rüdiger Daub

Fraunhofer-Institut für Gießerei-, Composite- und Verarbeitungstechnik IGCV
Am Technologiezentrum 10, 86159 Augsburg
www.igcv.fraunhofer.de