

Steuerungskonzept für virtualisierte und lernfähige Materialflusssysteme

A control concept for virtualized and learning-capable material flow systems

Anike Murrenhoff¹
Moritz Roidl²
Michael ten Hompel^{1,2}

¹ Fraunhofer-Institut für Materialfluss und Logistik
Dortmund

² Lehrstuhl für Förder- und Lagerwesen
Fakultät Maschinenbau
Technische Universität Dortmund

Automatisierte Logistiksysteme, die in unbekanntem Umgebungen unter weitgehendem Verzicht auf lokale Infrastruktur effizient arbeiten, erfordern neue Steuerungskonzepte für den Material- und Informationsfluss. Anhand der Implementierung eines Drohnen-schwarms wird in diesem Beitrag ein Konzept vorgestellt, das eine dezentrale Schwarmsteuerung mit Reinforcement Learning und einer Virtualisierungsarchitektur verbindet. Eine passende Entwicklungsumgebung für cyberphysische Systeme wird am Beispiel des Innovationslabors des Lehrstuhls FLW der TU Dortmund beschrieben.

[Schlüsselwörter: Cyberphysische Systeme, Schwarmsteuerung, Virtualisierung, verteilte Systeme, verteilte Intelligenz]

Automated logistics systems that operate efficiently in unknown environments without local infrastructure require new control concepts for the flow of materials and information. Based on the implementation of a drone swarm, this paper presents a concept that combines decentralized swarm control with reinforcement learning and a virtualization architecture. A suitable development environment for cyber-physical systems is described using the example of the Innovation Laboratory of the FLW Chair of the TU Dortmund University.

[Keywords: cyber-physical systems, swarm control, virtualization, distributed systems, distributed intelligence]

1 EINLEITUNG

Zentraler Kern dieses Beitrags ist eine Schwarmsteuerung für Systeme von Transportrobotern, die weitestgehend auf lokale, fest verbaute Infrastruktur verzichten. Ergänzend wird beschrieben, wie die Methode des Reinforcement Learnings den Schwarm modular um nicht-programmierte Verhaltensweisen ergänzen kann. Zusammen mit einer IT-Architektur auf Basis cyberphysischer Systeme und moderner Mobilfunktechnologien wird ein Steuerungskonzept entwickelt, das die Lernfähigkeit zum integralen Bestandteil der Materialflussteuerung macht.

Der Beitrag ist folgendermaßen gegliedert: Zu Beginn wird anhand von Schlüsseltechnologien motiviert, warum zukünftige Intralogistiksysteme ihre bislang üblichen Systemgrenzen verlieren werden und sich auch in unbekanntem Umgebungen selbst organisieren können. Anschließend wird auf die Grundlagen von dezentralen Steuerungen, maschinellen Lernverfahren und Virtualisierungstechniken in Bezug auf logistische Systeme eingegangen und insbesondere der Aspekt der Lernfähigkeit für Materialflussteuerungen betrachtet.

Der nächste Abschnitt beschreibt im Detail die Anpassung klassischer Schwarmalgorithmen auf die Steuerung von Transportrobotern und geht dabei auch auf spezifische Aufgaben wie eine Sequenzierung von Transportaufträgen und roboterspezifische Verhaltensweisen ein. Es wird das Konzept einer zellbasierten IT-Architektur beschrieben, bei der die Schwärme über 5G-Mobilfunktechnologie mit großen Rechenzentren verbunden werden, in denen anhand von Materialflusssimulationen mittels Reinforcement Learning laufend angepasste Betriebsstrategien entwickelt werden.

Zum Abschluss des Beitrags wird als Proof of Concept die Implementierung eines Drohnenschwarms im Forschungszentrum des Lehrstuhls für Förder- und Lagerwesen der TU Dortmund beschrieben. Dabei wird insbesondere auf die Soft- und Hardwarearchitektur der Entwicklungsumgebung eingegangen, die physische und virtuelle Objekte nahtlos miteinander verbindet. Ein Fazit beschreibt die Erfahrungen aus der Entwicklung des Drohnenschwarms und es wird ein Ausblick auf die Entwicklung eines neuartigen Transportroboterschwarms gegeben.

2 MOTIVATION UND STAND DER TECHNIK

Eine zentrale Herausforderung bei der Entwicklung automatisierter Intralogistiksysteme ist die Anforderung an die Steuerung, immer flexibler und dabei zugleich effizienter zu handeln – eine zentrale Ambivalenz, insbesondere für ein turbulentes Marktumfeld, das durch Komplexitätssteigerung und Dynamisierung geprägt ist (vgl. [HH17]). Ein Grund für die fehlende Flexibilität ist, dass die automatisierten Systeme durch menschliche Planer vorgeplant werden und nur auf Ereignisse und Situationen reagieren können, die in einer vorbestimmten und daher bekannten und begrenzten Umgebung entstehen und durch eine ortsfeste technische Infrastruktur gekennzeichnet sind.

Die Effizienz dieser vorgeplanten Systeme ist nur so lange gegeben, bis ungeplante Situationen und Ereignisse auftreten. Logistiksysteme sehen sich heutzutage jedoch hochdynamischen Märkten ausgeliefert, im Prinzip einem sich permanent verändernden Datenstrom von Kundenaufträgen, der durch seine schlechte analytische Vorhersagbarkeit zu einem dauerhaften Zustand ungeplanter Situationen führt (vgl. [HH17]).

Ein grundlegendes Systemkonzept, das mit einem dauerhaften Zustand von hochdynamischen Kundenauftragsströmen effektiv mithalten kann, muss auf die gleichen Systemarchitekturen aufsetzen, mit denen die Kunden ihre Aufträge generieren (vgl. Mobile Commerce), so dass eine systemweite, technologische Flexibilität auch in der Auftragserfüllung gewährleistet werden kann. Auf automatisierte Intralogistiksysteme übertragen ergibt sich die Forderung nach mobilen Systemarchitekturen, die weitgehend auf lokale, ortsfeste technische Infrastruktur verzichten können. Insbesondere bedeutet dies den Verzicht auf fest verkabelte Sensoren, ortsfest installierte Fördertechnik oder lokale, nicht-mobile Datenverarbeitung. Im Idealfall entfallen alle Umbautätigkeiten und entsprechend notwendige Umplanungsprozesse vollständig.

Gleichzeitig entsteht so ein System, das keine technisch bedingten räumlichen Grenzen hat: Durch das Fehlen fester Infrastruktur gibt es keine notwendig vorbestimmte Umgebung mehr, die einen ordnungsgemäßen und effizienten Prozessablauf garantiert. Die Verantwortung dafür liegt zukünftig vollständig bei intelligenten Ladehilfsmitteln und mobilen Transportrobotern.

Ein solches System bedarf einer deutlich erhöhten Selbstwahrnehmungsfähigkeit durch eine große Anzahl mobiler Sensoren. Deren Daten werden automatisiert zu einem laufend aktualisierten, virtuellen Modell verarbeitet und dienen als Grundlage für lernfähige Selbstorganisationsmechanismen, die menschliche Planungstätigkeiten weitgehend übernehmen.

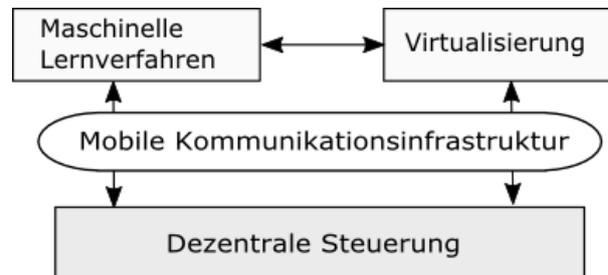


Abbildung 1. Zusammenhang der Schlüsseltechnologien für eine virtualisierte und lernfähige Materialflusssteuerung

Die tatsächliche Umsetzbarkeit eines solchen infrastrukturlosen, lernfähigen Intralogistiksystems basiert auf mehreren Schlüsseltechnologien, die ineinandergreifen (vgl. Abbildung 1). Es gibt zum einen die zunehmende Entwicklung cyberphysischer Systeme (CPS) und der damit einhergehenden dezentralen Steuerungsalgorithmen. Schwärme von Shuttles, Drohnen oder Fahrerlosen Transportsystemen [WHZ18] und sich selbst optimierende Sensoren [SHS18] oder Behälter, die bemerken, wenn sie bewegt werden und sich im Raum lokalisieren [MVT⁺18], sind physische Belege hierfür. Der konsequente Einsatz von CPS ermöglicht die erforderliche Erhöhung der Selbstwahrnehmung auf Basis zahlreicher mobiler Sensorik.

Die zweite Schlüsseltechnologie betrifft die Vernetzung und rechtzeitige Verarbeitung der Sensorik-Datenströme, die erst bei geeigneter Zusammenführung zur virtuellen Modellbildung dienen können. Die neuen Mobilfunktechnologien der fünften Generation (5G) ermöglichen genau eine solche ortsunabhängige Konsolidierung der Daten in den großen Rechenzentren der Cloud. Neben der Möglichkeit einer schnellen Übertragung großer Datenmengen haben 5G-Technologien den Vorteil, auch eine energieeffiziente Übertragung für stark ressourcenbeschränkte Objekte, wie etwa intelligente Ladehilfsmittel, bereitzustellen. Am Beispiel der intelligenten Palette zeigt sich, dass schon jetzt ein solches System ohne Batterietausch über Jahre hinweg weltweit Daten übertragen kann (vgl. [Gra19], [iPA18]). Es kann somit nicht nur auf eine Verkabelung verzichtet werden, sondern auch auf lokale Kommunikationsinfrastruktur, etwa WLAN-Router, so dass die Kommunikationsmöglichkeiten aus Sicht des Intralogistiksystems weit über die bisherigen Systemgrenzen hinausreichen.

Die echtzeitfähige Vernetzung auf Basis von 5G-Funknetzen hat auch zur Folge, dass CPS nicht mehr über eine große Rechenleistung vor Ort verfügen müssen, wie

etwa ortsfest installierte Server, sondern auf die vergleichsweise grenzenlose Rechenleistung der Cloud zurückgreifen können [CCT⁺18]. Hier kommen IT-Virtualisierungstechniken zum Einsatz, die eine Ausführung von Algorithmen auf geeignete Hardwareplattformen verteilen. Aus einer Erweiterung der Verteilung auf die Hardwareplattformen intelligenter physischer Objekte ergeben sich virtualisierte CPS. Dies wirkt sich auf zwei verschiedene Weisen aus: Zum einen kann jedes ressourcenbeschränkte Objekt prinzipiell auf große Rechen- und Speicherkapazitäten zurückgreifen, zum anderen lassen sich die dezentral verteilten Steuerungsalgorithmen der Objekte ins Rechenzentrum übertragen und in Simulationen einsetzen.

Für die konkrete technische Umsetzung des Lernens und Planens stellt die dritte Schlüsseltechnologie, die künstliche Intelligenz, geeignete Methoden und Werkzeuge zur Verfügung. Insbesondere maschinelle Lernverfahren können auf gespeicherte Sensordaten zurückgreifen und aus diesen Vergangenheitsdaten Schlüsse für aktuelle Betriebsstrategien ziehen. Über Simulationen virtueller Modelle können Verfahren eingesetzt werden, die aus spekulativen Zukunftsszenarien lernen und somit lernen, auch auf unvorhergesehene Situationen bestmöglich reagieren zu können. Bisher ist an genau dieser Stelle das Erfahrungswissen menschlicher Experten unabdingbar, die die Reaktion des Systems auf die Änderung der Umgebungseinflüsse abschätzen können. Es ist absehbar, dass diese Experten zukünftig die Aufgabe übernehmen, diese lernfähigen Verfahren anzulernen, anstatt wie bisher ihr Wissen direkt in das System zu übersetzen.

Durch einen kombinierten Einsatz dieser Schlüsseltechnologien wird eine neue Dimension verteilter Netze künstlicher Intelligenz auf physischen Objekten entstehen. Damit Intralogistiksysteme die Vorteile einer solchen Systemarchitektur nutzen können, bedarf es eines neuen Konzepts für die Architektur von Materialflusststeuerungen.

3 GRUNDLAGEN

Im den folgenden Unterabschnitten werden einige Grundlagen beschrieben, die für die Konzeptentwicklung in diesem Beitrag von Bedeutung sind. Es wird auf die Eignung von Schwarmalgorithmen für den Einsatz in unbekanntem Umgebungen eingegangen. Der Begriff der Lernfähigkeit für Materialflusssysteme wird erläutert sowie das Zusammenspiel von Reinforcement Learning und Virtualisierung.

3.1 DEZENTRALE STEUERUNG

In Reaktion auf die Problemstellung des sich hochdynamisch entwickelnden Marktumfelds wurden in den vergangenen Jahren flexible Intralogistiktechnologien entwickelt, deren maßgebliches Ziel eine schnellere, kostengünstigere Umplanbarkeit und Umbaufähigkeit ist.

Dies geschieht durch Modularisierung der Gewerke, die einzeln beim Hersteller entwickelt und schon vor der Installation am Einsatzort grundlegend getestet werden können (vgl. [GH10]). Charakteristisch ist, dass die Steuerungslogik am physischen Objekt in Form eines eingebetteten Rechnersystems lokale Entscheidungen trifft und durch Kommunikation mit anderen Modulen übergreifende Systemfunktionen abbildet.

Dezentrale Steuerungen für logistische Systeme werden seit einigen Jahren erfolgreich entwickelt und teilweise bereits industriell eingesetzt. Ein umfassender Überblick über dezentrale Steuerungsmethoden findet sich in [Sei16], einige Beispiele und Entwicklungen werden im Folgenden genannt.

Dezentrale Steuerungsalgorithmen können sowohl bei Stetig- als auch bei Unstetigfördertechnik zum Einsatz kommen. Bei dezentral gesteuerten Stetigfördersystemen entstehen ortsfeste Fördermodule, deren Layout sehr einfach umgebaut und ohne Programmierung geändert werden kann. Diese verbesserte Umplanungsfähigkeit findet weiterhin in den Systemgrenzen der lokalen technischen Infrastruktur statt. Bei dezentral gesteuerten Unstetigfördersystemen übernehmen die Aufgabe autonome Transportroboter. Der Einsatz intelligenter Ladehilfsmittel kann beide Systemtypen ergänzen (vgl. [ERB⁺12]).

Die Hauptaufgabe dezentraler Steuerungsalgorithmen besteht darin, die Transportaufträge durch lokale Entscheidungen der autonomen Einheiten zu steuern und zugleich die Sicherstellung der Bedienung aller Prozesse im System zu gewährleisten. Die übergreifende Systemkoordination wird durch ein Multiagentensystem gesteuert, das durch Einsatz standardisierter Kommunikationsprotokolle die lokalen Entscheidungen beeinflusst. Die Agentenprogramme müssen nicht notwendigerweise am physischen Objekt laufen, sondern können als Stellvertreter eines Objekts auch in anderen Laufzeitumgebungen ausgeführt werden. Die Verteilung der verschiedenen Agenten auf geeignete Hardwaresysteme und die Beherrschung der erhöhten Kommunikationstätigkeit im System gelten aktuell als Herausforderung bei der Entwicklung auf diesem Gebiet.

Die Entscheidungen der autonomen Einheiten in einem dezentral gesteuerten System führen zu einem komplexen, globalen Systemverhalten, in dem sich zunehmend nichtlineare, durch den menschlichen Betrachter kaum zu prognostizierende Zusammenhänge ergeben können. Den physischen Raum in Zellen aufzuteilen, in denen die direkte Interaktion zwischen Agenten stattfindet, ist eine Möglichkeit der Strukturierung. Zellübergreifende Kommunikation wird nicht direkt, sondern hierarchisch über eine verteilte Kommunikationsinfrastruktur durchgeführt. Eine zellbasierte Steuerungslogik wurde beispielsweise am Fraunhofer IML in Form von zellulärer Fördertechnik als

Konzept für Produktionssysteme bereits erfolgreich entwickelt (vgl. Abbildung 2).

Die folgenden zwei Ansätze zur dezentralen Koordination und Bewegungssteuerung werden in diesem Beitrag näher betrachtet:

- Topologische Graphen
- Schwarmalgorithmen

Bei der Modellierung von Intralogistiksystemen können Ressourcen und möglichen Transportwege als topologischer Graph abgebildet werden, dessen Knoten und Kanten die Speicherung relevanter Informationen für graphbasierte Algorithmen ermöglichen. Die Vorteile topologischer Graphen für Steuerungsalgorithmen liegen in der Vorhersagbarkeit der Bewegungen im Raum. So können auch Mensch und Maschine in relativer Sicherheit parallel arbeiten, solange sichergestellt ist, dass die Fahrwege frei bleiben. Die Koordination der Ressourcennutzung wird über die Reservierung von Knoten und Kanten gewährleistet. In diesem Zusammenhang eignen sich insbesondere zeitfensterbasierte Planungsalgorithmen, die Aufschluss über benötigte Ausführungszeiten geben und so eine genaue Sequenzierung oder Termineinhaltung ermöglichen.



Abbildung 2. Beispiel zellularer Fördertechnik am Fraunhofer IML in Dortmund

So wurde für den Flexsorter eine Logik für die dezentrale Steuerung von Stetigfördersystemen entwickelt, die auf dem Ansatz topologischer Graphen basiert (vgl. [Sei16]). Dabei bilden die physischen Module des Förderers selbst die Knoten des Graphen, die Übergänge zwischen den Modulen werden als Kanten abgebildet und Reservierungen der Ressourcen werden durch logische Zeit koordiniert. Topologische Graphen bei frei verfahrenen Unstetigfördersystemen wie FTF wurden am Fraunhofer IML eingesetzt. Dabei wird der topologische Graph auf den Fahrzeugen selbst gespeichert und Änderungen kommuniziert (vgl. [IML11]). Zeitfensterbezogene Steuerungsansätze, bei denen die Reservierungen der Ressourcen kontinuierlich aktualisiert werden, sind in [BLS⁺17] und [MWZ⁺10] zu finden.

Ein grundsätzlicher Nachteil der Nutzung von festen, topologischen Graphstrukturen für Transportroboter ist die Beschränkung der Bewegungen auf die Kanten des Graphen. Im Gegensatz dazu stehen Schwarmalgorithmen, bei denen die Bewegung nicht beschränkt wird und das Bewegungsverhalten emergent durch ein Zusammenspiel der Schwarmteilnehmer entsteht. Die einzelnen Verhaltensweisen bilden als modulare Bestandteile des Schwarmalgorithmus einzelne Ziele des Individuums ab (z. B. Kollisionsvermeidung, Exploration oder Bewegung zu einem Ziel). Die Koordination im Schwarm ergibt sich durch die Beobachtung der lokalen Umgebung, insbesondere anderer Schwarmteilnehmer, und die anschließende Reaktion durch Anpassung der Bewegungsrichtung oder Geschwindigkeit. Eine Besonderheit im Vergleich zu anderen dezentralen Steuerungsmethoden ist der Verzicht auf explizite Kommunikation bei der Koordination. Stattdessen geschieht dies durch die gegenseitige Beobachtung des Verhaltens der Teilnehmer. Schwarmalgorithmen wurden ursprünglich mit dem Ziel entwickelt, das natürliche Schwarm- oder Herdenverhalten von Tieren in der Natur nachzubilden (vgl. [Ray87]).

Im Kontext intralogistischer Systeme sind die einzelnen Verhaltensweisen des Schwarmalgorithmus vereinfacht mit Verkehrsregeln vergleichbar, die durch Beobachtung der lokalen Umgebung möglichst kommunikationsarm einen reibungsfreien Verkehrsablauf für einen Großteil der auftretenden Situationen im Normalbetrieb ermöglichen. Ein Vorteil der Nutzung von Schwarmalgorithmen in der Intralogistik ist, dass sie keine fest definierten Fahrwege benötigen, sondern im Regelfall alle zur Verfügung stehenden Freiflächen nutzen können. Dies begünstigt den Einsatz in unbekanntem Umgebungen und die Reaktion auf unvorhersehbare Ereignisse oder Störungen, da die Schwarmteilnehmer sich kontinuierlich und selbstständig auf neue Situationen einstellen können.

Ein weiterer Vorteil von Schwarmalgorithmen ist, dass durch ihren spezifischen, modularen Aufbau zusätzliche Verhaltensweisen einfach hinzugefügt werden können, ohne die bereits existierenden verändern zu müssen. Dies ermöglicht eine Anpassung an technische Besonderheiten und anwendungsfallspezifische Restriktionen, wie etwa nicht befahrbare Bereiche, Sicherheitsabstände oder technische Fahrzeugbeschränkungen.

Eine grundsätzliche Schwäche von Schwarmalgorithmen ist, dass sie als metaheuristische Verfahren stark von Layout, Lastzustand sowie Fahrzeugdichte und -typ beeinflusst sind. Bedingt durch die relative Nicht-Vorhersagbarkeit des emergenten Schwarmverhaltens ist eine Leistungsbestimmung solcher Systeme mit klassischen Berechnungsmethoden nicht möglich. Die Programmierung einer Schwarmsteuerung bedarf daher einer speziellen Entwicklungsumgebung, insbesondere einer geeigneten Simulation des Zusammenspiels der Schwarmteilnehmer.

Im weiteren Verlauf dieses Beitrags wird ein Schwarmalgorithmus zur Steuerung einer Fahrzeugflotte in einer hochdynamischen Systemumgebung vorgestellt, der für den Einsatz in der Logistik durch eine Einbindung eines topologischen Graphen ergänzt wird (vgl. Abschnitt 4). In diesem Zusammenhang wird zudem im nächsten Abschnitt das Konzept der Lernfähigkeit von logistischen Systemen erläutert.

3.2 LERNFÄHIGE LOGISTIKSYSTEME

Das Design geeigneter Systemarchitekturen stellt aktuell eine große Herausforderungen für einen allgemeinen, praktischen Einsatz maschineller Lernverfahren in industriellen Umgebungen dar (vgl. [Law19]). Es bedarf einer standardisierten Dekomposition der zu lösenden Probleme in der jeweiligen Anwendungsdomäne, einer Sicherstellung der Datenqualität schon an den datenerzeugenden Instanzen und einer Fähigkeit zum adaptiven, kontinuierlichen Lernen. Ein lernfähiges Logistiksystem muss diese Anforderungsfelder erfüllen um in der Lage zu sein, ohne explizite (Neu-)Programmierung auf dynamische Änderungen in einer teilweise unbekanntem, physischen Umgebung reagieren zu können.

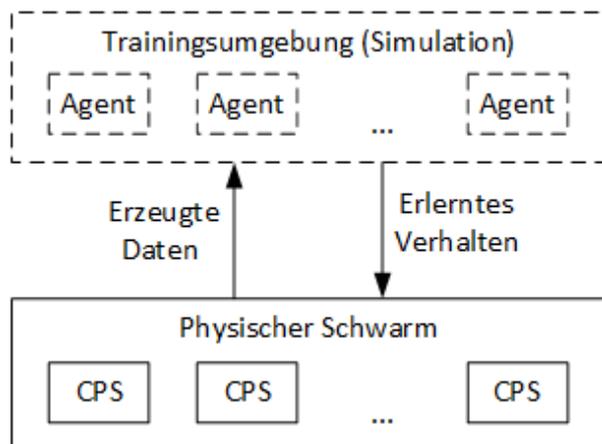


Abbildung 3. Zusammenhang zwischen physischem System und simulierter Trainingsumgebung bei Reinforcement Learning

Die Entwicklung dezentraler Steuerungsverfahren, wie sie in Abschnitt 3.1 beschrieben werden, hat in den letzten Jahren schon die Grundlage für eine standardisierte Dekomposition logistischer Systemaufgaben gelegt. Eine Orientierung an Verhaltensweisen aus der Natur findet sich, ähnlich wie bei Schwarmalgorithmen, auch in der Entwicklungsgeschichte des bestärkenden Lernens wieder (engl. Reinforcement Learning, vgl. [NNN18]). Beim Reinforcement Learning erlernen die Agenten eines Systems eine Verhaltensweise durch die Interaktion mit ihrer Umgebung. Basierend auf beobachteten Parametern der Umgebung führen sie Aktionen durch, die anschließend bewertet werden. Wünschenswerte Verhaltensweisen werden dabei belohnt und unerwünschte Verhaltensweisen bestraft

(vgl. [HKT18], [HG17]). Der Agent erstellt so eine Zuordnung von Beobachtungen zu Aktionen, die langfristig die erwartete Belohnung maximieren, und lernt aus Erfahrungen (vgl. u. a. [SB15]). Da ein Agent eine große Anzahl an Versuchen und Fehlversuchen zur Bestimmung von geeigneten Zuordnungen durchführen muss, benötigt er eine Trainingsumgebung, die Abläufe in den physischen Systemen simuliert (vgl. Abbildung 3).

Reinforcement Learning wurde bereits in verschiedenen Anwendungsbereichen in Kombination mit Schwarmalgorithmen angewandt, beispielsweise zum Erlernen eines Anti-Raubtier-Verhaltens (vgl. [MIN⁺06]), in Systemen, in denen mehrere Roboter gemeinsam ein Objekt in einer unbekanntem Umgebung bewegen (vgl. [MDK⁺18]), oder zur Steuerung unbemannter Luftfahrzeuge (vgl. [HG17]). Einen umfassenden Überblick über Anwendungen, Herausforderungen und Lösungen im Bereich Reinforcement Learning in Multi-Agenten Systemen und unter anderem für die Schwarmsteuerung bieten Nguyen et al. in [NNN18].

Mit der Entwicklung von Deep Reinforcement Learning wurde ein Durchbruch für den praktischen Einsatz in hochkomplexen Szenarien erzielt (vgl. [MKS⁺15]). Der Einsatz von Deep Reinforcement Learning bietet sich in der Entwicklung lernfähiger Materialflussteuerungen an, die ohne menschliche Unterstützung effektive Betriebsstrategien erlernen können. Die Rolle des menschlichen Experten wird sich dahingehend ändern, die Verhaltensweisen der Agenten im System über die richtigen Funktionen zu bewerten, sodass die wünschenswerte Verhaltensweise des Gesamtsystems schnellstmöglich erlernt wird (Human-in-the-Loop Machine Learning, vgl. [XML18⁺]).

Eine Systemarchitektur für eine lernfähige Materialflussteuerung wird in Zukunft durch folgende Eigenschaften und Fähigkeiten charakterisiert sein:

- Erzeugung qualitativ hochwertiger Datenströme für Reinforcement Learning in der Simulation (Data First Paradigm, vgl. [Law19])
- kontinuierliches Weiterlernen und Dazulernen während des Betriebs (Lifelong Machine Learning, vgl. [CL18])
- Automatisierte Überwachung des Einsatzes von Gelerntem im laufenden Betrieb (Hypervisor-Technologie, siehe auch Virtualisierung, vgl. [Law19])
- Einbindung von Expertenwissen ohne Programmierung (AutoML, vgl. [HZC19])

Eine dezentrale Steuerung auf Basis ressourcenbeschränkter, mobiler Hardwareumgebungen, wie sie in infrastrukturlosen CPS vorhanden sind, kann Verfahren des

maschinellen Lernens aufgrund der beschränkten Rechenleistung unter Umständen nicht vor Ort ausführen. Im folgenden Abschnitt wird daher auf die Virtualisierung als wichtige Eigenschaft für IT-Systemarchitekturen eingegangen, die in der Lage sind, eine Verteilung der Steuerung auf geeignete Hardwareressourcen durchzuführen (siehe auch Federated Learning, vgl. [BEG⁺19]).

3.2.1 VIRTUALISIERUNG

Unter dem Begriff der Virtualisierung wird im weitesten Sinne die Einführung logischer Abstraktionsschichten verstanden, die zwischen den physischen Ressourcen und der digitalen Außenrepräsentation des logistischen Gesamtsystems liegen. Gleichzeitig bedeutet Virtualisierung die Erweiterung jedes einzelnen physischen Objekts um seine digitalen Daten und insbesondere um die Datenströme, die es während seiner Lebenszeit im logistischen System erzeugt. Wie bereits in der Motivation beschrieben, wirkt sich dies auf zwei verschiedene Weisen aus: Zum einen kann jedes ressourcenbeschränkte Objekt auf große Rechen- und Speicherkapazitäten zurückgreifen, zum anderen lassen sich die dezentral verteilten Steuerungsalgorithmen der CPS ins Rechenzentrum übertragen und in Simulationen einsetzen.

Einen Überblick über den Stand der Technik im Bereich der Virtualisierung von Industriedaten im Allgemeinen bieten die Veröffentlichungen der Initiative International Dataspace (IDS) (vgl. [IDS19]). Der IDS definiert einen Standard für Datenströme, der unter anderem räumliche und zeitliche Informationen zu jedem erzeugten Datensatz sowie von Informationen über die Datenqualität (z. B. die Messgenauigkeit) beinhaltet.

Im Hinblick auf die Virtualisierung von Steuerungsalgorithmen sind nicht nur die Datenströme, sondern auch die Veröffentlichung und Verifizierung der einzelnen Softwaremodule sowie deren Verteilung auf physische Laufzeitumgebungen von Bedeutung. Eine Virtualisierung bedeutet in diesem Fall, dass es für die prinzipielle Funktionalität eines Softwaremoduls unerheblich ist, ob es auf der ressourcenbeschränkten Hardware eines eingebetteten Systems oder auf einem Server in der Cloud läuft. Dezentrale Steuerungsalgorithmen eignen sich durch ihre gegebene Modularität besonders für den Einsatz in virtualisierten Umgebungen.

Eine solchermaßen virtualisierte Softwarearchitektur bildet auch die Basis für eine virtuelle Abbildung des Materialflusssystems in der Simulation. So können Kopien der dezentralen Steuerungsalgorithmen parallel ausgeführt werden und zum Beispiel für Reinforcement Learning genutzt werden. Eine konsequente Umsetzung der Virtualisierung des Steuerungsverhaltens bildet die Basis für die Erzeugung sogenannter Digitaler Zwillinge.

Ein weiterer wichtiger Aspekt bei der Virtualisierung von Materialflusssystemen ist die Reaktionsfähigkeit

des Systems, die durch die Latenzzeit bei der Datenübertragung bestimmt wird. Softwaremodule werden nach Latenzzeiten auf die Schichten verteilt: Zeitkritische Module können näher am physischen Objekt laufen, während übergeordnete Module auf leistungsfähiger Hardware weiter entfernt platziert werden. Das Konzept des Edge Computings ermöglicht im Gegensatz zum Cloud Computing die Verteilung der Intelligenz an den Rand des Netzwerks (vgl. [Sat17]).

Virtualisierte Steuerungsalgorithmen müssen in der Lage sein, zur Laufzeit des Systems die Verteilung der Softwaremodule zu verändern, ohne die logistische Leistung des Systems zu beeinträchtigen. So kann bei einer Veränderung der äußeren Rahmenbedingungen, wenn etwa eine leistungsfähigere Kommunikationsinfrastruktur installiert wird, eine Anpassung automatisiert durchgeführt werden.

Viele KI-Algorithmen, insbesondere Verfahren des Deep Learning, benötigen große Datenmengen, um sinnvoll zu lernen (vgl. [ZYC⁺18]). Durch die Virtualisierung ergibt sich die folgende grundsätzliche Architektur: Lernen in der Cloud mit großer Rechenkapazität und anschließender Einsatz des Erlernten auf dem Shopfloor für schnelle, dezentrale und lokale Entscheidungen.

3.3 ZUSAMMENFASSUNG

Schwarmalgorithmen bieten eine gute Grundlage für die Entwicklung von Materialflusssystemen in hochdynamischen Umgebungen. Aufgrund der einfachen Erweiterung der Schwarmverhaltensweisen eignen sie sich gut für den Einsatz zusammen mit Reinforcement Learning. Beide Ansätze erfordern den Einsatz von Simulationsumgebungen in der Entwicklung, um das emergent entstehende Gesamtsystemverhalten validieren zu können. Der Einsatz von Virtualisierungstechniken hilft, das Sammeln und Analysieren von Daten außerhalb des physischen Logistiksystems zu ermöglichen und die Erkenntnisse aus der Simulation und den maschinellen Lernverfahren zurück in das System zu spielen. Ein virtualisiertes und lernfähiges Intralogistiksystem erhält so die Fähigkeit zur Selbstorganisation, und zudem kann es die Planungsergebnisse automatisch umsetzen, da durch das Fehlen einer lokalen, ortsfesten Infrastruktur Umbauarbeiten entfallen.

4 KONZEPT

Das in diesem Beitrag entwickelte Steuerungskonzept greift die beschriebenen Entwicklungen auf und basiert auf einer physischen Zellenstruktur, in der ein Schwarmalgorithmus Transportroboter steuert. Im Anschluss an die Vorstellung der physischen Rahmenbedingungen wird zunächst der Schwarmalgorithmus im Detail erläutert und eine Erweiterung um die Fähigkeit zur Sequenzierung von Transportbewegungen beschrieben. Zum Abschluss wird

auf die IT-Architektur zur Virtualisierung und Lernfähigkeit eingegangen.

4.1 RAHMENBEDINGUNGEN DER PHYSISCHEN SCHWARMUMGEBUNG

Die Systemgrenzen werden nicht durch ein starres Layout vorgegeben, sondern durch eine physische Zellstruktur bestimmt, die während des Betriebs veränderbar ist. Das Gesamtsystem besteht aus einer zusammenhängenden Anzahl von nebeneinanderliegenden Zellbereichen. Die jeweiligen physischen Abmessungen können je nach Implementierung auf verschiedene Weisen definiert werden: beispielsweise eine automatisiert erstellte Umgebungskarte der baulichen Gegebenheiten oder die Definition über eine zugeordnete Mobilfunkzelle. Die Umgebung des Schwarms wird durch die Grenzen der physischen Zelle bestimmt, innerhalb der sich der einzelne Schwarmteilnehmer selbst lokalisieren kann. Der physische Raum wird dreidimensional abgebildet und es wird von einer kontinuierlichen Zeitentwicklung ausgegangen.

Innerhalb einer Zelle begrenzen virtuelle Zäune die Freiflächen, die für Schwarmteilnehmer zur Verfügung stehen (siehe Abbildung 4). Diese können sich in den eingezäunten Bereichen frei bewegen und weichen Hindernissen und anderen Teilnehmern aus. Sie sind in der Lage, einen Zielpunkt zu verfolgen, der von außen vorgegeben wird. Dies kann sowohl ein statischer Zielpunkt sein (siehe Stern in Abbildung) als auch ein anderer Teilnehmer oder ein mobiles Hindernis in Bewegung.

Im Unterschied zu statischen, baulichen Hindernissen, die sich während des Schwarmbetriebs nicht bewegen, stellen mobile Hindernisse Menschen, einfache physische Objekte wie Ladehilfsmittel oder externe Robotersysteme dar, die nicht vom Schwarmverhalten gesteuert werden. Die Schwarmteilnehmer müssen die mobilen Objekte erkennen und ihnen dynamisch ausweichen können.

4.2 ANGEPASSTE SCHWARMSTEUERUNG FÜR LOGISTIKSYSTEME

Das in diesem Beitrag entwickelte Steuerungskonzept basiert auf den Elementen der bekannten Schwarmalgorithmen (vgl. [Ray87]), die an die Erfordernisse intralogistischer Systeme angepasst werden. Da klassische Schwarmalgorithmen das Verhalten von Tieren in der Natur abbilden, kann eine direkte Übertragung auf die Domäne Logistik nicht erfolgen. Ein übergeordnetes Transportverhalten steuert daher als Zustandsautomat das logistisch sinnvolle Verhalten der Roboter und aktiviert oder parametrisiert die einfachen, nicht veränderten Verhaltensweisen des Schwarmalgorithmus (siehe Abbildung 5). Die logistische Prozesssemantik wie Transportaufträge, Quellen, Senken und Ladungsträger ist im Transportverhalten isoliert gekapselt.

Das übergeordnete logistische Transportverhalten eines Teilnehmers greift auf die einfachen Verhaltensweisen zu, aktiviert oder deaktiviert sie und verändert dynamisch die Parameter und steuert so den logistischen Prozess.

4.2.1 ALLGEMEINE VERHALTENSWEISEN

Allgemeine Verhaltensweisen bilden die Basis der Schwarmsteuerung und werden von allen Schwarmteilnehmern berücksichtigt. Sie werden in individuelle und kollektive Verhaltensweisen unterteilt (siehe Abbildung 5). Individuelle Verhaltensweisen werden einzig durch ihre spezifischen Parameter und die physische Umgebung beeinflusst. Sie ermöglichen es einem Teilnehmer, individuelle Ziele zu erreichen. Die kollektiven Verhaltensweisen dienen der kommunikationsfreien Koordination unter den Teilnehmern und verhindern beispielsweise Kollisionen, gleichen die Bewegungsrichtung an oder halten Teilnehmer zusammen.

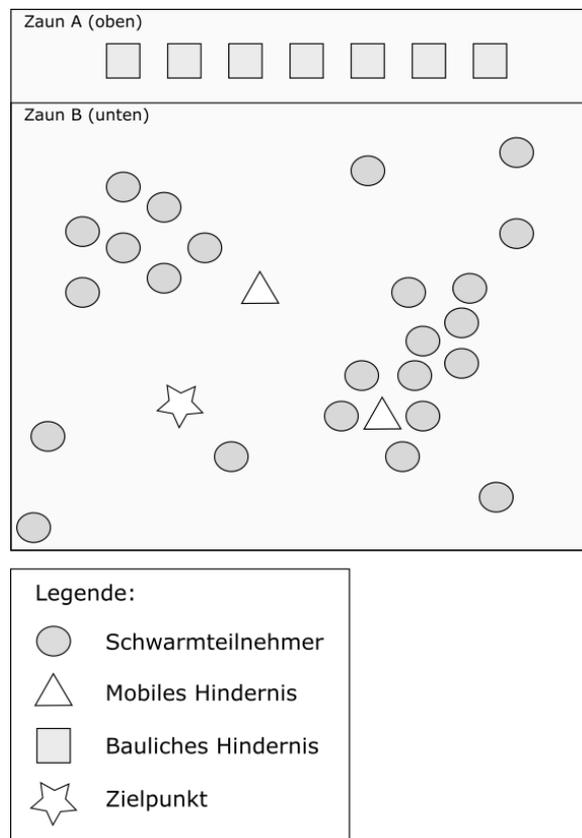


Abbildung 4. Schwarmumgebung und -elemente in der physischen Zelle

In den folgenden Abschnitten wird im Detail auf die allgemeinen Verhaltensweisen des Schwarmalgorithmus eingegangen und ihre Wirkung in logistischen Systemen erläutert.

Wandern: Das Wandern beeinflusst die Rotation eines Teilnehmers und ändert sie geringfügig in bestimmten Zeitintervallen um einen zufälligen Wert. Es entsteht eine

Art zielloser Bewegung im Raum, die zur Erkundung der Umgebung führt. Bei einer größeren Anzahl von Teilnehmern kann der Schwarm so in kurzer Zeit die gesamte Umgebung erkunden. Aus logistischer Sicht hat das Wandern keine direkte Funktion, jedoch kann es als Heuristik eingesetzt werden, welche die Teilnehmer gleichmäßig im Raum verteilt, um so schneller auf zufällig auftretende Transportaufträge reagieren zu können. Bei sich hochdynamisch verändernden Umgebungen liefert das Wandern einen Datenstrom, der für die Analyse der aktuellen Situation genutzt werden kann.

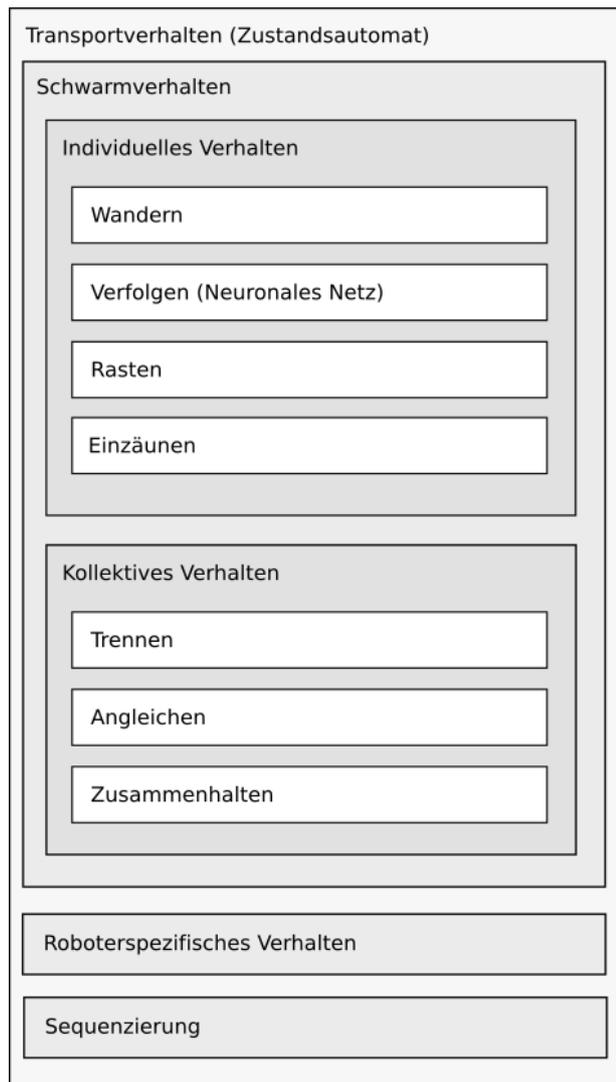


Abbildung 5. Struktur der Schwarmsteuerung für Materialflusssysteme

Verfolgen: Beim Verfolgen wendet sich der Teilnehmer einem selbstgewählten oder vorgegebenen Ziel zu und bewegt sich in dessen Richtung, bis er es erreicht. Ein logistischer Transportauftrag besteht aus dem Verfolgen der Quelle, der Lastaufnahme, dem Verfolgen der Senke und der Lastabgabe. Handelt es sich um ein bewegtes Ziel, kann beim Angleichen der Geschwindigkeit eine Bewegung in

Formation entstehen. Auf diese Weise können mehrere kleinere Teilnehmer beispielsweise eine große oder sperrige Last transportieren. Das Verfolgen ist das grundlegende Verhalten für die logistische Funktion. Daher wurde es im Rahmen der Implementierung zu diesem Beitrag exemplarisch per Reinforcement Learning angelernt (siehe Abschnitt 5).

Rasten: Im Gegensatz zum Wandern sucht sich der Teilnehmer beim Rasten einen Rastplatz, bewegt sich nicht mehr und wartet, bis er ein neues Ziel verfolgt. So werden der Energieverbrauch gesenkt und gleichzeitig Flächen für wichtige Transportaufträge freigehalten. Ebenso kann die Ladestation eines batteriegetriebenen Roboters als Rastplatz definiert werden. Stellt der Teilnehmer die Beobachtung der Umgebung ein und reagiert nicht mehr auf andere Teilnehmer in der Nähe, so wird er zu einem Hindernis. Alternativ dazu kann er auch beim Rasten auf andere Teilnehmer reagieren und ihnen den Weg freimachen.

Einzäunen: Das Einzäunen begrenzt die Bewegung eines Teilnehmers auf einen spezifischen physischen Bereich. Jeder Teilnehmer ist mit einer aktuellen Referenzeinzäunung assoziiert. Er prüft während der Bewegung, ob der Zaun erreicht ist, und wendet sich in diesem Fall dem Mittelpunkt des eingezäunten Bereichs zu. Das Einzäunen überschreibt üblicherweise Vorgaben anderer Verhaltensweisen. Wird zum Beispiel ein Zielpunkt außerhalb des Zauns verfolgt, dann bleibt der Teilnehmer innerhalb der Einzäunung. Im Kontext der zellbasierten Aufteilung des physischen Raums werden die Zellgrenzen durch Zäune realisiert. Ein Zellwechsel geschieht so über den Wechsel des Referenzzauns. Hier zeigt sich der Vorteil von überlappenden Zellen, da sich ein Teilnehmer beim Wechsel niemals außerhalb des jeweiligen Referenzzauns wiederfinden kann.

Trennen: Das Trennverhalten gehört zu den kollektiven Verhaltensweisen und vermeidet Kollisionen zwischen Teilnehmern des Schwarms oder Hindernissen. Es berechnet die Richtungsvektoren zu anderen Teilnehmern und Hindernissen in der näheren Umgebung, bildet den invertierten, durchschnittlichen Richtungsvektor und wendet den Teilnehmer so von allen möglichen Kollisionsrichtungen ab. Da jeder Schwarmteilnehmer dieses Verhalten im Kollektiv zeigt, ergibt sich während der Laufzeit ein Mindestabstand zwischen den Teilnehmern, der eingehalten wird. Damit das Trennverhalten auch im realen Robotersystem Kollisionen vermeiden kann, müssen der Mindestabstand und die Geschwindigkeit so gewählt werden, dass die beteiligten Roboter rechtzeitig Richtungsänderungen durchführen können. Bei Robotern, die sich sehr schnell bewegen oder anderweitig träge bei Richtungsänderungen reagieren, führt dies zu sehr großen Mindestabständen oder zur Begrenzung der zulässigen Maximalgeschwindigkeit im Schwarmbetrieb. Eine Lösung für diese Problematik bieten zeitenfensterbasierte Reservierungen, wie sie in Abschnitt 4.2.3 beschrieben werden.

Angleichen und Zusammenhalten: Die Verhaltensweisen des Angleichens und des Zusammenhaltens erzeugen bei klassischen Schwarmalgorithmen das typische Verhalten von Vogelschwärmen oder Herdentieren. Sie berechnen dabei den durchschnittlichen Richtungsvektor beziehungsweise die durchschnittliche Position der benachbarten Teilnehmer, so dass sich der Schwarm gemeinsam in enger Formation in eine Richtung bewegt. Im logistischen Anwendungsszenario finden diese beiden Verhaltensweisen in seltenen Fällen Anwendung. So kann bspw. ein größerer Kundenauftrag, der von mehreren Teilnehmern bearbeitet wird, zusammengehalten werden.

4.2.2 ZUSAMMENSPIEL DER ALLGEMEINEN VERHALTENSWEISEN IM LOGISTISCHEN SYSTEM

Die beschriebenen Verhaltensweisen bestimmen die Bewegungsvorgaben für jeden Teilnehmer des Schwarms durch ein individuelles Zusammenspiel basierend auf der lokalen Umgebung des einzelnen Roboters. Das Zusammenspiel zwischen den individuellen und relativ einfachen Verhaltensweisen erzeugt eine emergente Schwarmintelligenz. Diese kann durch eine Reihe von Parametern beeinflusst werden: die Reihenfolge, in der die Verhaltensweisen ausgeführt werden, der Grad des Einflusses, den das einzelne Verhalten auf die geplante Geschwindigkeit und Rotation eines Teilnehmers hat, und verhaltensspezifische Parameter wie die minimale oder maximale Entfernung zu den zu berücksichtigenden Nachbarn.

(1) ANWENDUNGSFALLSPEZIFISCHES VERHALTEN

Ein Schwarm, der bestimmte Aufgaben in einer Anwendung erfüllen soll, wird um ein übergeordnetes Verhalten ergänzt, das die allgemeinen Verhaltensweisen zur Laufzeit aktiviert, deaktiviert und parametrisiert. Diesem übergeordneten Verhalten obliegt es, die gestellte Aufgabe zu erfüllen, mit übergeordneten Systemen zu kommunizieren und spezifische Aktionen für die Anwendung auszulösen. In der Logistik wird diese Aufgabe von einem Transportverhalten übernommen (siehe Abbildung 5). Die Leistung eines solchen aufgabenorientierten Schwarms hängt stark von der Bestimmung und laufenden Anpassung der optimalen Parameter für den Anwendungsfall ab. Dabei sind im Zusammenspiel Regeln und Besonderheiten zu beachten, die im Folgenden erläutert werden.

Wandern, Verfolgen und Rasten sind sich gegenseitig ausschließende Verhaltensweisen, die von einem übergeordneten Verhalten aktiviert und deaktiviert werden. Entweder erkundet ein Teilnehmer frei, rastet oder folgt einem bestimmten Ziel. Das Einzäunen hält den Schwarm in einem bestimmten Bereich und ist zusammen mit dem Trennen wichtig für einen sicheren Betrieb. Schwarmteilnehmer nehmen Menschen als mobile Hindernisse wahr und weichen ihnen über das Trennverhalten aus, so dass eine sichere Mensch-Maschine-Interaktion gewährleistet ist und gleichzeitig der Schwarm weiterhin seine logistische Funktion erfüllen kann (kein Not-Aus). Ein gutes Beispiel

für das Zusammenspiel mehrerer Verhaltensweisen ergibt sich, wenn ein Schwarm auf einen Zaun trifft. Hier wenden sich die Teilnehmer an der Vorderseite des Zauns in Richtung des Mittelpunkts, während sich die Nachfolger aufgrund des Trennens oder des Angleichens an ihren Vorgängern orientieren. So dreht sich die Bewegungsrichtung des gesamten Schwarms im Kollektiv.

(2) ROBOTERSPEZIFISCHES VERHALTEN

Roboter, die als Teilnehmer des Schwarms agieren, müssen die Parameter der allgemeinen Verhaltensweisen auf ihre physischen Fähigkeiten abstimmen und können die Schwarmsteuerung um eigene Verhaltensweisen ergänzen. Zum Beispiel muss eine gelandete Drohne vor einer Positionsänderung starten und anschließend wieder landen, bevor ihre Batterieladung erschöpft ist. Eine Drohne wird die allgemeinen Verhaltensweisen daher um ein spezifisches Start- und Landeverhalten ergänzen. Während fliegende Drohnen ihre Flugrichtung ohne Wendemanöver direkt ändern können, muss ein typischer radgebundener Roboter erst eine vergleichsweise langsame Drehung einleiten. Grundsätzlich gilt, dass alle Roboter, die am Schwarm teilnehmen, die allgemeinen Verhaltensweisen adäquat umsetzen müssen, so dass das Gesamtsystem korrekt funktioniert. Die Schwarmsteuerung mit ihren standardisierten allgemeinen Verhaltensweisen bildet so eine logische Abstraktionsschicht in virtualisierten Logistiksystemen, da die eingesetzten Robotersysteme einfach austauschbar sind und verschiedene Roboter mit unterschiedlichen Fähigkeiten kollektiv Aufgaben erfüllen können.

4.2.3 ERWEITERUNG UM SEQUENZIERUNG

Die Schwarmsteuerung kann mit ihrer prinzipiell heuristischen Vorgehensweise keine Garantien für die Einhaltung von zeitlich terminierten Transportaufträgen oder für die Sequenzierung von Aufträgen bieten. Diese Fähigkeit kann durch die Einbindung topologischer Graphstrukturen ergänzt werden, auf denen zeitfensterbasierte Reservierungen durchgeführt werden (vgl. Abschnitt 3.1).

Die Knoten des topologischen Graphen werden als virtuelle Hindernisse mit Reservierungsagenten als temporäre Elemente der Schwarmumgebung eingeführt (siehe Abbildung 6). Die Kanten werden als Übergänge zwischen den virtuellen Hindernissen im jeweiligen Reservierungsagenten gespeichert. Die so entstehende Kette von Hindernissen wird von den Schwarmteilnehmern aufgrund des Trennverhaltens gemieden und so freigehalten.

Ein Schwarmteilnehmer (a) kann nun eine zeitfensterbasierte Reservierung durchführen, wobei sich die Zeiten nach der möglichen Geschwindigkeit des Roboters richten. Nach der Durchquerung eines virtuellen Hindernisses wird dieses gelöscht und der belegte Raum für den Schwarm freigegeben. Ein Roboter kann so eine zeitlich optimale Trajektorie planen, den benötigten Weg über die temporären, virtuellen Hindernisse reservieren und mit maximaler

Geschwindigkeit durchfahren. Durch die Vorausberechnung der Zeitfenster wird die Ankunftszeit bestimmbar und über die Reservierung garantiert. Dies bildet die Grundlage für echtzeitfähige Transporte.

Wenn sich zwei reservierte Ketten von virtuellen Hindernissen kreuzen (wie bei a und b in der Abbildung), werden diese verschmolzen und die Reihenfolge der Zeitfenster vom Reservierungsmanager (x) festgelegt. Über entsprechende Algorithmen, wie sie in graphbasierten, dezentralen Steuerungen Anwendung finden, kann ein deadlockfreies Routing auch in komplexen Graphstrukturen gewährleistet werden (vgl. [Sei16]).

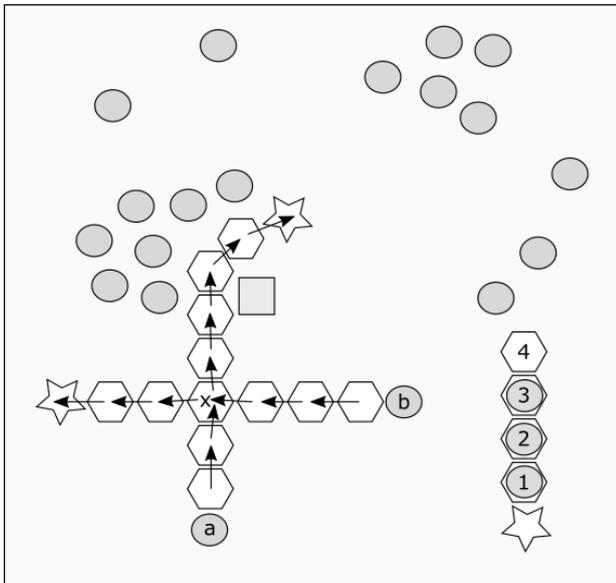


Abbildung 6. Einbindung topologischer Graphstrukturen für zeitfensterbasierte Reservierungen

Die Einhaltung einer vorgegebenen Sequenz von Transportaufträgen kann über eine Kette von virtuellen Hindernissen erreicht werden, die entsprechend der Sequenz reserviert wurden (siehe in Abbildung 6 die virtuellen Hindernisse 1, 2, 3 und 4). Den Transportaufträgen wird gemäß ihrer Sequenznummer das zugeordnete virtuelle Hindernis als Zwischenziel zur Pufferung vorgegeben. Erst wenn alle vorrangigen Aufträge durchgeführt wurden, darf das eigentliche Ziel angesteuert werden.

Diese Art der Einbindung topologischer Graphstrukturen kann auch für Bereiche einer Umgebung angewendet werden, in denen nicht genug Freifläche für ein gut funktionierendes Schwarmverhalten zur Verfügung steht (z. B. enge Regalgassen). In diesen Fällen werden die virtuellen Hindernisse permanent eingesetzt und können auch die Funktion von Einbahnstraßen übernehmen.

Grundsätzlich ist die Ergänzung um topologische Graphstrukturen notwendig, damit die Schwarmsteuerung in typischen logistischen Umgebungen eingesetzt werden

kann. Dabei erhöht sich jedoch die Variabilität und Komplexität des Verfahrens erheblich. Insbesondere stellt sich die Frage, inwiefern permanente topologische Graphstrukturen in Sinne einer Selbststeuerung automatisch erzeugt werden können und welches Verhältnis von Echtzeittransporten und Schwarmtransporten zu einer den vorgegebenen Kriterien entsprechenden optimalen Gesamtsystemleistung beiträgt.

4.3 VERTEILUNG DER INTELLIGENZ IN VIRTUALISIERTEN UMGEBUNGEN UND LERNFÄHIGKEIT

Wie in Abschnitt 3.2.1 beschrieben, kann die Intelligenz in virtualisierten cyberphysischen Systemen an verschiedenen Stellen verteilt werden. Die Lernfähigkeit des Systems besteht darin, dass die Schwarmteilnehmer Daten aus der physischen Umgebung zur Ermittlung einer optimalen Verhaltensweise in einer Trainingsumgebung zur Verfügung stellen und anschließend das erlernte, zurückgelieferte Verhalten umsetzen.

Abbildung 7 zeigt den Kontext als Ökosystem der Datenverarbeitungsschichten in Zusammenhang mit der Aufteilung des physischen Raums. Die globale Datenverarbeitung findet auf leistungsfähigen, virtualisierten Servern in der Cloud statt. Hier befinden sich übergeordnete Systeme zur Auftragssteuerung. Das digitale Abbild mit der logischen Abstraktionsschicht des virtualisierten Logistiksystems wird hier externen Systemen als Dienst angeboten. Anspruchsvolle Lernaufgaben wie das Reinforcement Learning auf Systemebene können auf spezialisierten Systemen in der Cloud effizient bearbeitet werden, während einfache Lernaufgaben auch in weiter unterhalb liegenden Ebenen gelöst werden.

Die ortsnahe Datenverarbeitung findet auf Mobilfunkstationen statt, die den Server und die Middleware klassischer Intralogistiksysteme ersetzen. Die Zellen des physischen Raums werden anhand der Zellen des Mobilfunknetzwerks festgelegt. Ein intelligentes Objekt, das sich in eine Mobilfunkzelle einwählt, wird dem dazu korrespondierenden Raum zugeordnet. Die allgemeine Softwarearchitektur ist als Multiagentensystem ausgelegt, dessen Agenten zur Laufzeit zwischen den verschiedenen Datenverarbeitungsebenen wechseln können. Die Agenten kommunizieren über standardisierte Protokolle. Neben den so entstehenden anwendungsspezifischen Datenströmen werden Metainformationen zur Leistung der Kommunikationsinfrastruktur bereitgestellt.

Das Steuerungskonzept bezieht sich auf alle intelligenten physischen Objekte einer Zelle, deren Agenten miteinander kommunizieren. Die Agenten können am physischen Objekt oder in der Mobilfunkzelle als Stellvertreter laufen. Entsprechend dem Konzept der virtualisierten Logistiksysteme wird ein von der genauen technischen Ausführung und Fähigkeit der Ladehilfsmittel und Robotersysteme abstrahiertes Modell gewählt. Dabei wird davon

ausgegangen, dass sich alle Roboter ausreichend genau lokalisieren können, die Umgebung beobachten und Hindernisse sowie andere Roboter in der näheren Umgebung erkennen können.

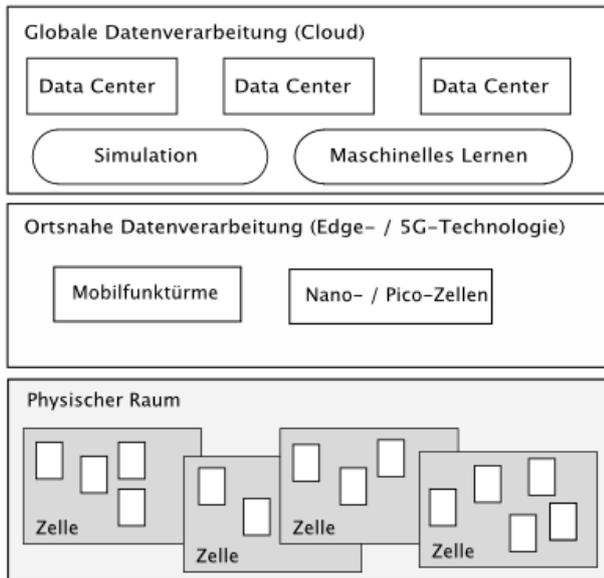


Abbildung 7. Datenverarbeitungsarchitektur in Zusammenhang mit einer zellbasierten Aufteilung des physischen Raums

Die Akteure im physischen logistischen System erzeugen Datenströme, beispielsweise eine laufende Aktualisierung der Position. Diese Daten werden zum Lernen der optimalen Verhaltensweise im System genutzt und in eine Simulationsumgebung gespeist. Aufgrund der beschränkten Rechen- und Akkuleistung findet das Lernen nicht auf den CPS selbst statt, sondern in der virtualisierten Umgebung. Dort sind die Akteure des physischen Schwarms als Digitale Zwillinge abgebildet, denen die Daten des physischen Systems zur Verfügung gestellt werden. Die Digitalen Zwillinge erlernen das beste Verhalten über Reinforcement Learning in der virtualisierten Umgebung auf Basis der Datengrundlage aus dem physischen System. Das erlernte Wissen wird in einem neuronalen Netz abgebildet, das anschließend zurück zu den Teilnehmern des Schwarms gespielt wird.

Auf diese Art und Weise können die einzelnen Verhaltensweisen der Schwarmsteuerung erlernt und durch neuronale Netze ersetzt werden, ohne dass sie explizit programmiert werden. Umgesetzt wurde dieses Prinzip bereits für die Verhaltensweise *Verfolgen* (vgl. Abschnitt 4.2.1).

5 IMPLEMENTIERUNG

Eine Referenzimplementierung des Steuerungskonzepts wurde im Forschungszentrum des Innovationslabors für hybride Dienstleistungen entwickelt, das sich in der Versuchshalle des Lehrstuhls für Förder- und Lagerwesen

an der TU Dortmund befindet (vgl. [Inn19]). Die Arbeiten zur Lernfähigkeit wurden in Zusammenarbeit mit dem Research-Clan Maschinelles Lernen aus dem Leistungszentrum Logistik und IT durchgeführt (vgl. [Lei19]). Die spezifische Architektur und Ausstattung des Forschungszentrums sind wesentliche Faktoren bei der erfolgreichen Entwicklung und werden daher im Folgenden beschrieben.

Das Forschungszentrum ist mit einer Infrastruktur ausgestattet, die für das Rapid Prototyping von CPS konzipiert wurde. Es befindet sich in einer Leichtbauhalle, die baugleich mit herkömmlichen Industriegebäuden im Logistikbereich ist und folgt dem Grundkonzept eines hochflexiblen Entwicklungslabors, dessen Versuchsfläche frei von fest installierten Gewerken ist. Die Halle ist mit mehreren Beobachtungssystemen ausgestattet, die an der Decke, an den Wänden und im Boden installiert sind. Im Rahmen der Implementierung wurden insbesondere das Motion Capturing System (MoCap) und ein Laserprojektionssystem verwendet.

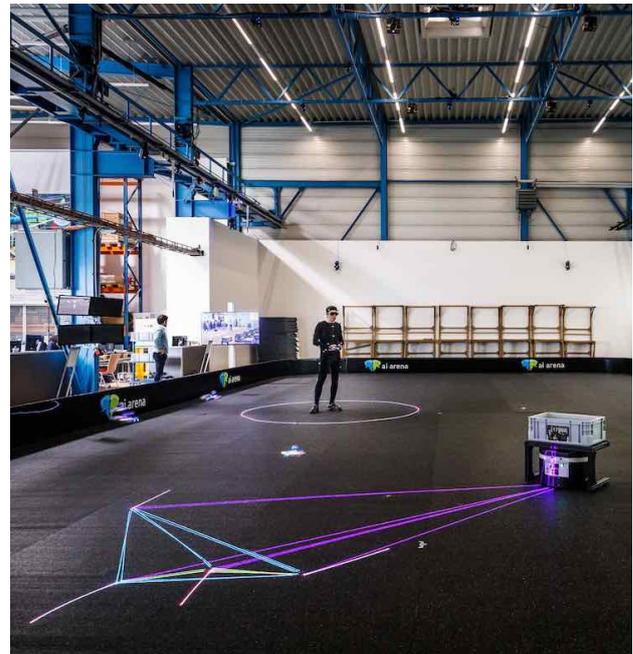


Abbildung 8. Gesamtansicht des Forschungszentrums während des Betriebs

Das optische MoCap-System basiert auf 40 Infrarotkameras und kann eine große Zahl von entsprechend markierten Objekten bei einer Genauigkeit von ca. 0,3 mm verfolgen. Es arbeitet mit einer Datenübertragungsrate von bis zu 300 Hz bei Latenzen von 4-15 ms. Der beobachtete Versuchsraum ist 22 m lang, 15 m breit und 4 m hoch. Die Lokalisierungsdaten sind über das Netzwerk für mehrere Clients gleichzeitig zugänglich. Neben einfachen physischen Objekten stehen mehrere Marker-Anzüge zur Personenortung zur Verfügung, deren Daten als komplexe, digitale Skelettmodelle bereitgestellt werden.

Acht Laserprojektoren decken den Versuchsraum ab und können farbige Vektorgrafiken auf den Boden projizieren.

Abbildung 8 zeigt eine typische Nutzung des Systems. Die Person, der Roboter und der Behälter werden vom MoCap-System beobachtet. Im Vordergrund projiziert das Lasersystem eine visuelle Darstellung der aktuellen internen Datenstrukturen der Robotersteuerung. Der projizierte Kreis um die Person im Hintergrund stellt den Sicherheitsbereich dar, den der Roboter nicht befahren darf. Der Behälter auf dem Roboter enthält ein eingebettetes System (schwarzes Quadrat auf der Vorderseite), das mit dem Roboter für die Durchführung der Transportaufgabe kommuniziert.

5.1 ENTWICKLUNGSUMGEBUNG

Abbildung 9 zeigt die aktuelle Entwicklungsarchitektur, die im Forschungszentrum verwendet wird. Alle markierten Objekte im physischen Raum werden über eine aktive MoCap-Verbindung in eine 3D-Simulationsumgebung (Unity 3D) gespiegelt. Neben der Darstellung physischer Objekte kann die Simulation beliebig viele virtuelle Objekte (z. B. Datenstrukturen der Algorithmen) enthalten.

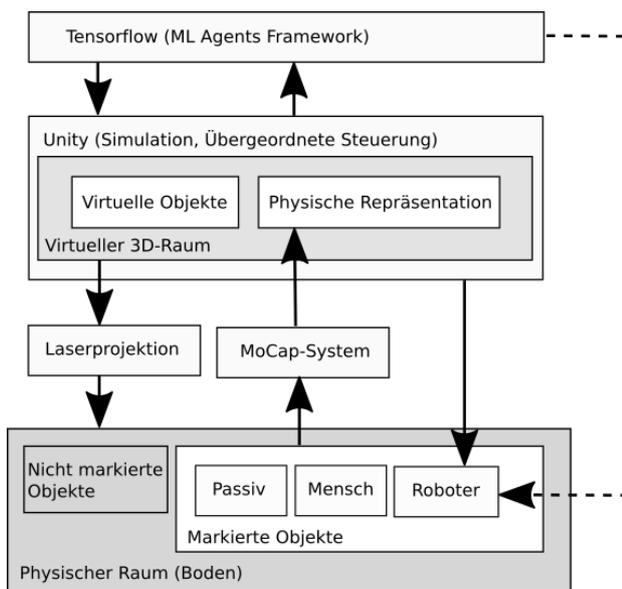


Abbildung 9. Architektur der Entwicklungsumgebung

Zu Beginn der Entwicklung befindet sich die gesamte übergeordnete Steuerungslogik zentral im Simulationswerkzeug, das im Steuerungsmodus einfache Bewegungsanweisungen an die Robotersysteme sendet und resultierende Objektbewegungen über die MoCap-Verbindung erhält. Virtuelle Objekte, die keine physische Repräsentation haben, werden über das Laserprojektionssystem auf dem Hallenboden gespiegelt. Befindet sich das Werkzeug im reinen Simulationsmodus, können Roboter- und Objektbewegungen zeitlich beschleunigt simuliert werden.

Die Entwicklung maschineller Lernverfahren geschieht über die Einbindung von Tensorflow, das über das ML-Agents-Framework an die Simulationsumgebung gekoppelt wird ([JBV⁺18]). Das Framework wurde für den Einsatz von Reinforcement Learning entwickelt und verwendet Unity als Trainingsumgebung. Über eine Python API können aktuelle Zustände des Systems, der Handlungsraum der Agenten und der Wert der Rewardfunktion aus der Simulationsumgebung mit Algorithmen in Tensorflow verknüpft werden. Über Proximal Policy Optimization (PPO) wird ein neuronales Netz antrainiert, das den aktuellen Zustand der Simulationsumgebung zu wünschenswertem Verhalten der Agenten kodiert. Während des Trainings wird die Simulation bis zu hundertfach beschleunigt ausgeführt. Das gelernte Verhalten in Form von neuronalen Netzwerken kann anschließend zur Steuerung in Unity verwendet werden.

In späteren Entwicklungsphasen wird die übergeordnete Steuerungslogik schrittweise vom Simulationswerkzeug auf die realen Laufzeitsysteme übertragen. Dies geschieht derzeit manuell durch Neuimplementierung in einer anderen Programmiersprache. Als Beispiel verwendet das aktuelle System C# in der Unity-Simulation, Python und C++ innerhalb der Roboter- und Drohnensteuerung basierend auf dem Robot Operating System (ROS) und C auf den eingebetteten Systemen der intelligenten Behälter und Drohnen. Eine automatische Übertragung der neuronalen Netzwerke per Codegenerierung auf die Laufzeitumgebungen befindet sich derzeit in der Entwicklung.

Die Architektur des hier entwickelten Steuerungskonzepts spiegelt sich in der Architektur der Entwicklungsumgebung des Forschungszentrums wider (vgl. Abbildung 7). Der physische Versuchsraum entspricht genau einer physischen Zelle, das Simulationswerkzeug im Steuerungsmodus korrespondiert mit der Laufzeitumgebung in der 5G-Zelle (Edge Computing), und die Simulationsumgebung im Simulationsmodus mit gekoppeltem Tensorflow entspricht ML-Frameworks, die auf leistungsfähiger Hardware in der Cloud laufen.

5.2 DROHNENSCHWARM

Zur Erprobung der Schwarmsteuerung wurde ein Demonstrator entwickelt, der alle wesentlichen Elemente des Verfahrens beinhaltet. Als Ziel bei der Implementierung der Schwarmsteuerung wurde ein Drohnenschwarm verwendet, der innerhalb der beschriebenen Entwicklungsumgebung des Forschungszentrums in Betrieb genommen wurde. Er besteht aus 16 Drohnen, die über eine erweiterte Version des Open-Source-Projekts CrazySwarm angesteuert werden.

Der Drohnenschwarm fliegt in einem per Laserprojektion angezeigten Bereich (Zaun) der Versuchsfläche. Versuchspersonen können den Bereich betreten, wenn sie über Stirnbänder mit Markern verfügen, so dass sie vom

Schwarm als mobile Hindernisse erkannt werden. Menschen können sich so sicher im Schwarm bewegen.

Über eine markierte Frisbeescheibe können durch die Versuchspersonen Transportaufträge generiert werden (siehe Abbildung 10). Das Ziel aller Aufträge ist ein per Laser angezeigter Bereich, der als Nest gekennzeichnet ist. Der Transportauftrag wird als virtuelles Objekt erzeugt und per Laser projiziert. Die Drohne mit dem kürzesten Weg zur Quelle nimmt das virtuelle Lastobjekt auf. Anschließend bringt sie die Last zum Nest und gliedert sich wieder in den Schwarm ein.



Abbildung 10. Drohnenschwarm im Forschungszentrum

Bei der Entwicklung des Demonstrators zeigte sich, dass mehrere Drohnen, die denselben Zielpunkt verfolgen (z. B. bei Abgabe der Last im Nest), sich durch das Trennverhalten gegenseitig dauerhaft blockieren können. Zur Behebung des Problems wurde das Transportverhalten erweitert, so dass es den Fortschritt bei der Verfolgung eines Ziels beobachtet. Sollte sich über längere Zeit die Distanz zum Zielpunkt nicht mehr verringern, wird die Drohne für ein zufällig gewähltes, kurzes Zeitintervall in das Wanderverhalten geschaltet. Dies führt bei mehreren Drohnen zu einer stochastisch bestimmten Sequenzierung und sorgt durch das Wandern für einen neuen Anflugvektor zum Ziel. Im Rahmen zahlreicher Vorführungen des Demonstrators konnte dieser Ansatz validiert werden. Diese heuris-

tische Methode der Blockadeauflösung kann in Fällen angewendet werden, bei denen genug Freifläche zum Ausweichen verfügbar ist und keine strikten zeitlichen Anforderungen bestehen.

Die prototypische Implementierung des Konzepts in der physischen Welt dient nicht nur als Proof-of-Concept. Die erfolgreiche Umsetzung für ein so komplexes Problem im dreidimensionalen Raum motiviert darüber hinaus weitere Anwendungsfälle, z. B. in der Steuerung von neuartigen Transportrobotersystemen.

6 FAZIT UND AUSBLICK

Dieser Beitrag zeigt, wie auf Basis aktueller technischer Entwicklungen hochdynamische Materialflusssysteme gesteuert werden, die innerhalb kürzester Zeit in Betrieb genommen werden können. Ein Grundprinzip bildet dabei der Verzicht auf eine fest verbaute, lokale Infrastruktur, sowohl für die physischen Materialflussprozesse, als auch für den Informationsfluss. Die Anbindung an moderne Telekommunikationsinfrastruktur ermöglicht eine Vielzahl hochdynamischer, mobiler Logistikressourcen in Echtzeit mit einer Recheneinheit in der Cloud (oder Edge) zu steuern und die dort verfügbaren Rechenkapazitäten auch zum Einsatz maschineller Lernverfahren zu nutzen.

Einige offene Fragestellungen für den Einsatz von Reinforcement Learning in der Schwarmsteuerung von Intralogistiksystemen wurden während der Entwicklung des Drohnenschwarms identifiziert und werden im Folgenden kurz beschrieben.

Die Bestimmung des optimalen Grads für den Einsatz der freien Schwarmsteuerung gegenüber einer zeitenfensterbasierten Planung mit topologischen Graphen ist eine Aufgabe, die im laufenden Betrieb durch die Fahrzeugflotte gelöst werden muss.

Ebenso kann die Prognose zeitlicher Abläufe einen Ansatzpunkt darstellen. Bei zeitenfensterbasierten Steuerungsverfahren kann eine reine Berechnung des Reservierungszeitfensters über Geschwindigkeit sowie Beschleunigung der Fahrzeuge zu ungenau sein.

Ein großes Potenzial scheint im Einsatz von Reinforcement Learning beim Erlernen spezifischer Verkehrsregeln und Betriebsheuristiken zu liegen, die das System auf die Eigenheiten eines spezifischen Einsatzortes anpassen.

Die Auswahl günstiger Rastplätze könnte über maschinelle Lernverfahren bestimmt werden, die virtuelle Kosten für Zonen in der Umgebung erzeugen, so dass gute Rastplätze (in der Nähe von Quellen) belohnt und schlechte (an Kreuzungspunkten) bestraft werden.

Eine weitere wesentliche Fragestellung bei dezentral gesteuerten Systemen ist die der Verteilung der Intelligenz im System. Dabei wird die geeignete Trennung von lokaler, autonomer Intelligenz und kollaborativer, globaler Planung untersucht. Die erhöhte Komplexität der Aufgabe in großen logistischen Systemen ist ein Ansatzpunkt für den Einsatz maschineller Lernverfahren.

Im Allgemeinen hat sich gezeigt, dass maschinelle Lernverfahren, Virtualisierung und dezentrale Steuerung Themen sind, die bei der Entwicklung logistischer Systeme bislang relativ unabhängig voneinander betrachtet werden. Dieser Beitrag zeigt die Vorteile eines integrativen Ansatzes zur Vereinigung dieser drei Bereiche. Dezentrale Steuerungsalgorithmen, insbesondere Schwarmalgorithmen, bilden durch ihre inhärente Modularität eine geeignete Basis für einen standardisierten Einsatz in unternehmensübergreifenden logistischen Netzwerken. Die steigende Zahl von intelligenten, physischen Objekten sorgt in Zukunft für eine immer größere Anzahl von Datenströmen und damit für eine erhöhte Qualität und Aktualität der virtuellen Simulationsmodelle, die für den Einsatz maschineller Lernverfahren zur Verfügung stehen. Eine Herausforderung besteht darin, das gelernte Wissen in die dezentrale Steuerung zu integrieren.



Abbildung 11. Einsatz von Schwarmverfahren am Beispiel des Loadrunners

Das vorgestellte Steuerungskonzept zeigt, dass ein schwarmbasierter Ansatz gut geeignet ist für neue logistische Anforderungen. Derzeit befindet sich mit dem Loadrunner eine neue Generation von Transportrobotern in der Entwicklung, die im Forschungszentrum des FLW erprobt und das hier beschriebene Steuerungskonzept einsetzen werden (siehe Abbildung 11).

In Zukunft ist geplant, die unterschiedlichen Forschungshallen des Lehrstuhls und des Fraunhofer-Institutes um eine gemeinsame 5G-Installation zu ergänzen und mit einem bereits existierenden Hochleistungscluster für Maschinelles Lernen zu verbinden. Somit werden größer angelegte Feldstudien zur Validierung des Konzepts und die robuste Weiterentwicklung der zugrundeliegenden Algorithmen ermöglicht.

7 FÖRDERHINWEIS

Diese Veröffentlichung wurde im Rahmen der Forschungsprojekte „Innovationslabor Hybride Dienstleistungen in der Logistik“ vom Bundesministerium für Bildung und Forschung (BMBF) und „Leistungszentrum Logistik und IT“ mit Mitteln der Fraunhofer-Gesellschaft und des Landes Nordrhein-Westfalen gefördert.

LITERATUR

- [BEG⁺19] K. Bonawitz, H. Eichner, W. Grieska, D. Huba, A. Ingermann, u. a. „Towards Federated Learning at Scale: System Design“, arXiv, 2019, <http://arxiv.org/abs/1902.01046>
- [BLS⁺17] C. Blesing, D. Luensch, J. Stenzel, B. Korth, „Concept of a Multi-agent Based Decentralized Production System for the Automotive Industry“, in Advances in Practical Applications of Cyber-Physical Multi-Agent Systems, (PAAMS) 15th International Conference Proceedings, 2017.
- [CCT⁺18] J. Cheng, W. Chen, F. Tao, und C.-L. Lin, „Industrial IoT in 5G environment towards smart manufacturing“, Journal of Industrial Information Integration, Bd. 10, S. 10–19, Juni 2018.
- [CL18] Z. Chen, B. Liu, „Lifelong Machine Learning (Second Edition)“, in Synthesis Lectures on Artificial Intelligence and Machine Learning, Morgan & Claypool, 2018, <https://www.morganclaypool.com/doi/abs/10.2200/S00832ED1V01Y201802AIM037>
- [ERB⁺12] J. Emmerich, M. Roidl, T. Bich, M. ten Hompel, „Entwicklung von energieautarken, intelligenten Ladehilfsmitteln am Beispiel des inBin“, in Logistics Journal, Vol. 2012, 2012
- [Gra19] A. Granzow, „Die Palette der Zukunft ist interaktiv“, in Deutsche Verkehrs-Zeitung, DVV, Hamburg, 29.11.2019, <https://www.dvz.de/rubriken/management-mema-recht/paletten-und-verpackung/detail/news/die-palette-der-zukunft-ist-interaktiv.html>, aufgerufen am 02.12.2019
- [GH10] W. Günthner, M. ten Hompel (Hrsg.), „Internet der Dinge in der Intralogistik“, Springer, Berlin, 2010

- [HH17] M. ten Hompel, M. Henke, „Logistik 4.0 – Ein Ausblick auf die Planung und das Management der zukünftigen Logistik vor dem Hintergrund der vierten industriellen Revolution“, in Handbuch Industrie 4.0 Bd.4: Allgemeine Grundlagen, Springer, Berlin, S. 249-259, 2017
- [HKT18] P. Hernandez-Leal, B. Kartal, und M. E. Taylor, „Is multiagent deep reinforcement learning the answer or the question? A brief survey“, *learning*, 2018, 21. Jg., S. 22.
- [HZC19] X. He, K. Zhao, X. Chu, „AutoML: A Survey of the State-of-the-Art“, *arXiv*, 2019, <https://arxiv.org/abs/1908.00709>
- [IDS19] International Data Spaces Association, „Reference Architecture Model“, Version 3.0, April 2019, <https://www.internationaldataspaces.org/publications/reference-architecture-model-3-0/>, aufgerufen am 14.07.2019
- [IML11] Fraunhofer IML, „Fraunhofer IML eröffnet neue Forschungshalle für Zelluläre Fördertechnik: Dortmunder Wissenschaftler lernen von Ameisen“, *Pressemitteilung* 22.06.2011, https://www.iml.fraunhofer.de/de/presse_medien/pressemitteilungen/zfthalle.html, aufgerufen am 14.07.2019
- [Inn19] Innovationslabor Hybride Dienstleistungen in der Logistik, <https://www.innovationslabor-logistik.de/forschungszentrum/>, aufgerufen am 14.07.2019
- [iPA18] iPAL - Intelligente Palette, Fraunhofer-Institut für Materialfluss und Logistik IML, und European Pallet Association e.V. (EPAL), „LogiMAT-Weltpremiere: Fraunhofer IML und EPAL heben Datengold der Logistik“, *Pressemitteilung* 2018, <https://ipal-pallets.org/de/Home/Press>, aufgerufen am 14.07.2019
- [JBV⁺18] A. Juliani, V. Berges, E. Vckay, Y. Gao, H. Henry, M. Mattar, D. Lange, „Unity: A General Platform for Intelligent Agents“, *arXiv preprint arXiv:1809.02627*, <https://github.com/Unity-Technologies/ml-agents>, 2018.
- [Law19] D. Lawrence, „Data Science and Digital Systems: The 3Ds of Machine Learning Systems Design“, in *Data Science and Digital Systems*, 26.03.2019, <http://arxiv.org/abs/1903.11241>, aufgerufen am 14.07.2019
- [Lei19] Leistungszentrum Logistik und IT, <https://leistungszentrum-logistik-it.de/>, aufgerufen am 14.07.2019
- [MDK⁺18] S. V. Manko, S. A. K. Diane, A. E. Krivoshatkiy, I. D. Margolin, und E. A. Slepynina, „Adaptive Control of a Multi-Robot System for Transportation of Large-Sized Objects Based on Reinforcement Learning“, 2018 IEEE Conference of Russian Young Researchers in Electrical and Electronic Engineering (EIConRus), S. 923–927, 2018.
- [MIN⁺06] K. Morihiro, T. Isokawa, H. Nishimura, und N. Matsui, „Emergence of Flocking Behavior Based on Reinforcement Learning“, in B. Gabrys, R.J. Howlett, und L.C. Jain (Hrsg.) „KES 2006“, Part III, LNAI 4253, S. 699–706, 2006.
- [MKS⁺15] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, S. Petersen, „Human-level control through deep reinforcement learning“, in *Nature*, 518(7540), S. 529-533, 2015
- [MVT⁺18] M. Masoudinejad, A. K. R. Venkatapathy, D. Tondorf, D. Heinrich, R. Falkenberg, und M. Buschhoff, „Machine Learning Based Indoor Localisation Using Environmental Data in PhyNetLab Warehouse“, in *Smart SysTech 2018; European Conference on Smart Objects, Systems and Technologies*, 2018, S. 1–8.
- [MWZ⁺10] A. Mors, C. Witteveen, J. Zutt, F. Kuipers, „Context-Aware Route Planning“, 8th German Conference, MATES 2010, 2010.
- [NNN18] T. Nguyen, N. Nguyen, und S. Nahavandi, „Deep Reinforcement Learning für Multi-Agent Systems: A Review of Challenges, Solutions and Applications“, *arXiv preprint arXiv:1812.11794*, 2018.
- [Pla18] Plattform Industrie 4.0, „MCP- Mobile Controlled Production / 5G for Digital Factories“, Working Paper, Technology Scenario v1.1, 19.03.2018,

<https://www.plattform-i40.de/PI40/Redaktion/EN/Downloads/Publikation/mobil-gesteuerte-produktion.pdf>, aufgerufen am 14.07.2019

- [Ray87] C. W. Reynolds, „Flocks, Herds, and Schools: A Distributed Behavioral Model”, Computer Graphics, Bd. 21, Nr. 4, S. 25-34, 1987.
- [Sat17] M. Satyanarayanan, „The Emergence of Edge Computing”, Computer, Bd. 50, Nr. 1, S. 359–371, Januar 2017.
- [SB18] R. B. Sutton, A. Barton, „Reinforcement Learning. An Introduction”, The MIT Press, Cambridge, MA, London, Second Edition, 2018.
- [Sei16] Z. Seibold, „Logical Time for Decentralized Control of Material Handling Systems”, Wissenschaftliche Berichte Institut für Fördertechnik und Logistiksysteme am Karlsruher Institut für Technologie (KIT), Bd. 89, 2016.
- [SHS18] A. Schütze, N. Helwig, T. Schneider, „Sensors 4.0 – smart sensors and measurement technology enable Industry 4.0“, J. Sens. Sens. Syst., Bd. 7, Nr. 1, S. 359–371, Mai 2018.
- [SST17] Y. Sun, T. Su, Z. Tu, „Faster R-CNN Based Autonomous Navigation for Vehicles in Warehouse”, 2017 IEEE International Conference on Advanced Intelligent Mechatronics (AIM), 2017.
- [WHZ18] J. Wen, L. He, und F. Zhu, „Swarm Robotics Control and Communications: Imminent Challenges for Next Generation Smart Logistics“, IEEE Communications Magazine, Bd. 56, Nr. 7, S. 102–107, Juli 2018.
- [XML+18] D. Xin, L. Ma, J. Liu, S. Macke, S. Song, A. Parameswaran, „Accelerating Human-in-the-loop Machine Learning“, in Proc. Of the Second Workshop on Data Management for End-to-End Machine Learning, 2018, <https://doi.org/10.1145/3209889.3209897>

Anike Murrenhoff, M.Sc., MSIE, Research Assistant at the Fraunhofer-Institute for Material Flow and Logistics, Dortmund.

Phone: +49 231 9743 202,
E-Mail: anike.murrenhoff@iml.fraunhofer.de

Dipl.-Inform. Moritz Roidl, Chief Engineer at the Chair of Materials Handling and Warehousing, Technical University Dortmund.

Phone: +49 231 755-3092,
E-Mail: moritz.roidl@tu-dortmund.de

Prof. Dr. Dr. h. c. Michael ten Hompel, Head of the Chair of Materials Handling and Warehousing, Technical University of Dortmund and Managing Director of the Fraunhofer-Institute for Material Flow and Logistics.

Phone: +49 231 9743-600,
E-Mail: michael.tenHompel@tu-dortmund.de

Address: Joseph-von-Fraunhofer-Str. 2-4, 44227 Dortmund, Germany