

Kooperierende Mobile Roboter

Cooperating Mobile Robots

Sven Behnke, Universität Bonn

Der Einsatz von Teams kooperierender mobiler Roboter ist erforderlich, wenn eine Aufgabe inhärent verteilt ist oder nicht von einem Roboter allein bewältigt werden kann. Auch wenn Aufgaben prinzipiell von Einzelrobotern erledigt werden können, kann der Einsatz mehrerer Roboter Vorteile bringen, da Roboterteams schneller, robuster und skalierbarer sein können. Bei der Steuerung kooperierender mobiler Roboter müssen vielfältige Ressourcenzuordnungsprobleme gelöst werden, beispielsweise die Aufgabenzuordnung oder die Zuordnung von Gebieten des Arbeitsraums. Im Artikel werden Kommunikations- und Kontrollarchitekturen, sowie Pfadplanungs- und Lernverfahren diskutiert, die für Multi-Roboter-Systeme entwickelt wurden. Die mögliche Nützlichkeit kooperierender mobiler Roboter wird durch vier Anwendungsbeispiele illustriert.

The use of teams of cooperating mobile robots is necessary, if a task is inherently distributed or cannot be solved by a single robot. In cases when a task can in principle be completed by a single robot, the use of multiple robots can result in advantages in execution speed, robustness, and scalability. Multiple resource allocation problems must be solved to make robots cooperate, like task allocation or the allocation of portions of the work space. This article discusses architectures for communication and control in multi-robot systems as well as path planning and multi-robot learning. The possible utility of cooperating mobile robots is illustrated using four application examples.

Schlagwörter: Mobile Roboter, Kooperation, Multi-Agenten-Systeme.

Keywords: Mobile robots, cooperation, multi-agent-systems.

1 Motivation

Mobile Roboter werden seit vielen Jahren in der Forschung untersucht und inzwischen in vielen Anwendungen – zu Land, zu und unter Wasser und in der Luft – erfolgreich eingesetzt. Die Aufgaben dieser Roboter umfassen dabei vor allem Tätigkeiten, die Menschen nicht möglich wären (z.B. wegen fehlender Zugänglichkeit), die für Menschen zu gefährlich sind (z.B. Arbeiten in strahlenbelasteter Umgebung) oder die einfach und repetitiv sind (z.B. Transportaufgaben in einer Werkhalle). Auch wenn Einzelroboter für viele Einsatzzwecke ausreichen, gibt es häufig Gründe, die den Einsatz mehrerer Roboter erforderlich machen oder zumindest vorteilhaft erscheinen lassen [41].

Der Einsatz mehrerer Roboter ist erforderlich, wenn ein Roboter allein nicht in der Lage ist, eine bestimmte Aufgabe zu erfüllen. Dies ist beispielsweise dann der Fall,

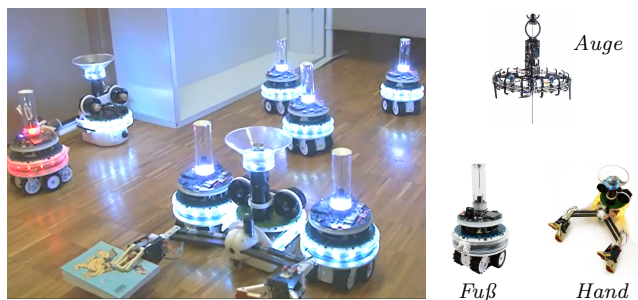


Bild 1: Aufgabenteilung im Swarmanoid-Projekt [11].

wenn ein zu transportierendes Objekt zu groß für einen einzelnen Roboter ist. Hier müssen mehrere Roboter das Objekt von verschiedenen Seiten halten und sich koordiniert bewegen. Ein anderes Beispiel für notwendige Kooperation mehrerer Roboter sind Spezialroboter, die jeweils nur Teilaufgaben lösen können, wie Roboter, die

mit spezieller Sensorik einen Defekt in einem Abwasserkanal finden, welcher dann von einem anderen Roboter repariert wird, oder die Augen-, Fuß- und Handroboter im Swarmanoid-Projekt (Abb. 1 [11]).

Der Einsatz mehrerer Roboter ist auch dann nötig, wenn die Aufgabe inhärent in einer Umgebung verteilt ist. Beispiele dafür sind die Überwachung eines großen Gebietes, das aufgrund von begrenzter Sensorreichweite und Verdeckungen nicht von einem einzelnen Roboter beobachtet werden kann oder auch das Aufspüren von Eindringlingen in einem Gebäude, die sich vor einem einzelnen Roboter verstecken könnten, indem sie in bereits untersuchte Gebiete wechseln wenn sie vom Roboter aus nicht sichtbar sind.

Auch wenn ein einzelner Roboter im Prinzip eine Aufgabe allein lösen kann, bringt die Verteilung auf mehrere Roboter häufig Vorteile. Beispielsweise kann eine Erkundungsaufgabe durch den parallelen Einsatz mehrerer Roboter schneller erledigt werden. Auch ist es häufig einfacher, mehrere Roboter mit beschränkten Fähigkeiten zu entwickeln als einen Superroboter, der eine Aufgabe allein löst. Ein Beispiel dafür sind Mannschaftsspiele, wie Fußball. Natürlich wäre ein Roboter denkbar, der den Ball immer sieht, schneller dribbelt als alle anderen Spieler ohne Ball rennen können und dann auch noch von beliebiger Position den Ball direkt in eine der oberen Torecken schießt. Allerdings ist der Bau eines solchen Einzelspielers ebenso wenig realistisch wie der Gewinn eines Bundesligaspiels durch nur einen Spieler. Um erfolgreich zu sein, müssen die Spieler einer Mannschaft sich den Ball zupassen, da nur so der Ball schneller bewegt werden kann als die gegnerischen Spieler rennen können.

Ein weiterer Aspekt ist die mögliche erhöhte Robustheit von Mehrrobotersystemen. Wenn mehrere Roboter sich gegenseitig bei der Erledigung von Teilaufgaben vertreten können und es keine einzelne Fehlerursache mehr gibt, die zum Systemversagen führt, wird das Gesamtsystem durch diese Redundanz robuster. Wenn allerdings für die Erledigung einer Aufgabe mehrere Spezialroboter benötigt werden, sinkt die Systemzuverlässigkeit.

Schließlich wird durch die Möglichkeit, die Kapazität eines Mehrrobotersystems durch einfaches Hinzufügen oder Entfernen von Robotern anzupassen eine leichte Skalierbarkeit der Systemleistung erzielt, beispielsweise um mit dem Wachstum eines Geschäftes mitzuhalten oder um zeitlich schwankenden Anforderungen zu entsprechen.

Selbstverständlich ist der Einsatz mehrerer Roboter auch mit Herausforderungen verbunden. Beispielsweise kann die Wahrscheinlichkeit des Auftretens eines Defekts bei mehreren preiswerten Robotern höher sein als bei einem teureren Qualitätsroboter. Es ist daher schwierig, die vollständige Verfügbarkeit eines Mehrrobotersystems sicher zu stellen. Auch stellt die Kontrolle

mehrerer Roboter deutlich höhere Anforderungen als die Kontrolle eines einzelnen Roboters.

Während ein einzelner Roboter in vielen Anwendungen, wie beispielsweise bei der Untersuchung gefährlicher Gegenstände oder der Telepräsenz, von einem menschlichen Experten teleoperiert werden kann, würde die gleichzeitige Steuerung mehrerer Roboter die menschliche Aufmerksamkeit überfordern. Hier sind eine abstrakte Steuerung ganzer Robotergruppen, beispielsweise beim Formationsflug oder beim Konvoifahren, oder zumindest teilautonomes Verhalten der einzelnen Roboter, beispielsweise bei Transportaufgaben im Krankenhaus, erforderlich.

Während bei stationären Robotern das Koordinationsproblem für mehrere Roboterarme, die beispielsweise in der Automobilproduktion zusammen arbeiten, im Vorfeld gelöst und dann wiederholt genutzt werden kann, ist bei mobilen Robotern die Situation in der Regel jeweils einzigartig und es muss das Koordinierungsproblem daher jeweils neu gelöst werden. In beiden Fällen müssen Ressourcenzuordnungsprobleme gelöst werden. Welcher Roboter arbeitet wann und wo an welcher Teilaufgabe? Dies führt typischerweise zu Ressourcenkonflikten, die aufgelöst werden müssen, um gemeinsam die jeweilige Aufgabe zu erfüllen. Andererseits sind bei geschickter Problemdekomposition viele Problemstellungen aus der lokalen Sicht eines einzelnen Roboters wieder einfach, was die Programmierung erleichtert.

Ähnliche Fragestellungen der Problemdekomposition und Lösungsfusion werden außerhalb der Robotik im Bereich der verteilten künstlichen Intelligenz, insbesondere im Gebiet der Multi-Agenten-Systeme, untersucht. Nach einer Diskussion dieser Ansätze geht der Artikel auf Roboterschwärme, Kommunikations- und Kontrollarchitekturen, Aufgabenzuordnung, Pfadplanung und Lernen ein. Zum Abschluss werden vier Anwendungsbeispiele vorgestellt: Versandlogistik, Konstruktion von Strukturen durch Flugroboter, Exploration und Roboterfußball. Nicht Gegenstand dieses Artikels sind zelluläre oder modulare Roboter, die aus mehreren kleineren Modulen einen rekonfigurierbaren Roboterkörper bilden, und stationäre Multi-Roboter-Systeme.

2 Multi-Agenten-Systeme

Das Gebiet der Multi-Agenten-Systeme (MAS) ist ein Teilgebiet der Künstliche-Intelligenz-Forschung, das sich mit Prinzipien für die Konstruktion von Systemen beschäftigt, in denen mehrere Agenten handeln – insbesondere wird die Koordination des Agentenverhaltens untersucht [50, 58]. Hierbei wird unter einem Agent ein autonom Handelnder verstanden, der in eine Umgebung eingebettet ist und eigene Ziele verfolgt. Beispiele für betrachtete Agenten sind interagierende Softwareprogramme, simulierte interagierende Roboter und

auch Gruppen realer Roboter. Die Umgebung wird dabei vom Agent über Sensoren wahrgenommen und durch Aktuatoren beeinflusst. Anwendungen von MAS reichen von Kommunikationsnetzwerken über automatisierte Beschaffungssysteme bis zu künstlichen Intelligenzen für Spielfiguren.

MAS sind sowohl für die Modellierung kooperativer Szenarien, in denen alle Agenten gemeinsame Ziele verfolgen, kompetitiver Szenarien, in denen die Ziele von Agentengruppen gegensätzlich sind, als auch allgemeinerer Szenarien, in denen einzelne Agenten eigene Ziele verfolgen und temporär Koalitionen bilden können, geeignet.

Eine weitere Dimension der Modellierung von MAS ist die Heterogenität der Agenten. Es können sowohl identische Agenten als auch verschiedenartige Agenten betrachtet werden, wozu auch identische Agenten mit unterschiedlichem Verhalten gezählt werden können.

Entscheidend für die Komplexität von Multi-Agenten-Systemen ist, ob und in welcher Intensität die Agenten kommunizieren. Wenn die Agenten nur implizit durch Interaktion mit einer gemeinsamen Umgebung kommunizieren, können nur einfache Verhaltensweisen entstehen, die häufig durch biologische Vorbilder inspiriert sind, wie Schwarmverhalten durch Interaktion benachbarter Agenten (Vorbild: Fische, Vögel) [13] oder Wegoptimierung durch Legen von Duftspuren (Vorbild: Ameisen) [10]. Wenn die Agenten nicht nur über die Umwelt, sondern auch direkt miteinander kommunizieren, können sie ihre Handlungen koordinieren und beispielsweise Rollen in einem Team übernehmen. Je mehr Informationen die Agenten austauschen, desto eher können sie optimale Problemlösungen finden. Beispielsweise können Agenten lokale Wahrnehmungen austauschen und durch deren Fusion globalere Sichten auf ihre Umgebung erzeugen und so mehr Kontext für ihre Entscheidungen berücksichtigen.

Im Extremfall der totalen Transparenz gibt es eine zentrale Instanz, die Zugriff auf alle sensorischen Messungen hat und alle Agenten dirigieren kann. Hierbei handelt es sich wieder um einen einzelnen Agenten, der nun Entscheidungen in einem sehr hochdimensionalen Zustands-Aktions-Raum treffen muss. Da die Komplexität von Planungs- und Lernverfahren typischerweise exponentiell mit der Dimensionalität des Problemraums steigt, ist dieser zentralistische Ansatz nur für Offline-Planung oder sehr kleine Agentengruppen praktikabel.

Für die Modellierung von MAS bietet sich häufig eine Formulierung als Markovsches Entscheidungsproblem an [43]. Hierbei werden Zustandsübergänge $X \times U \rightarrow X$ durch ihre Wahrscheinlichkeit $p(x_{t+1}|x_t, u_t)$ modelliert und eine skalare Belohnungs- oder Kostenfunktion $\rho(x_i, u_j)$ gibt den Erwartungswert der Belohnung für Aktion u_j in Zustand x_i an. Ziel eines rationalen Agenten ist es nun, den Erwartungswert R der Summe aller zukünftigen Belohnungen zu maximieren. Dazu wird

häufig eine Nutzenfunktion $Q(x, u)$ geschätzt. Wenn diese optimal ist, d.h. durch eine optimale Strategie der Aktionsauswahl erzeugt wurde, maximiert die Wahl der lokal vielversprechendsten Aktion $u^* = \arg \max_u Q^*(x, u)$ auch den insgesamt erwarteten Nutzen beim Start vom Zustand x .

Durch das gleichzeitige Handeln mehrerer Agenten in einer Umgebung kann häufig nicht angenommen werden, dass diese stationär ist. Insbesondere wenn die Agenten ihr Verhalten durch Lernen anpassen, ergeben sich dynamische Effekte welche die Anwendung klassischer Lernverfahren erschweren und beispielsweise zu einem „Wettrüsten“ zwischen Agentengruppen führen können.

Hierbei spielt eine Rolle, wie stark ein Agent das Wissen und Verhalten anderer Agenten modelliert. Unter der Voraussetzung rational handelnder Agenten, die perfektes Wissen über die anderen Agenten haben, erklärt die Spieltheorie die Existenz von Gleichgewichtszuständen, in denen keiner der Agenten einen Vorteil durch einseitige Änderung seines Handelns erzielen kann [21]. Dies kann beispielsweise für das Finden kooperativer Strategien benutzt werden, welche so subtil sein können wie das Rechts- oder Linksfahrgebot im Straßenverkehr, das die Wahrscheinlichkeit von Kollisionen gegenüber willkürlicher Fahrbahnwahl deutlich vermindert. Solche sozialen Konventionen reduzieren die Entscheidungsproblematik – rechts oder links dem Gegenverkehr ausweichen – deutlich. Eine weitere Möglichkeit, die möglichen Aktionen eines Agenten zu beschränken und so Entscheidungen zu vereinfachen, ist die Zuweisung von Rollen [59]. Beispielsweise ist es im Fußball nicht sinnvoll wenn alle Spieler gleichzeitig zum Ball laufen. Daher kann durch Rollenzuweisung (Torwart, Verteidiger, Stürmer, ...) eine Zuständigkeit je nach Spielsituation vorbestimmt werden. Die nicht zum Ball laufenden Spieler übernehmen dann andere Aufgaben.

Eine weitere Methode zur Komplexitätsreduktion sind Koordinationsgraphen [19]. Die Idee hierbei ist, die globale, hochdimensionale Nutzenfunktion $Q(x, u)$ als Summe lokaler Nutzenfunktionen Q_k darzustellen, die jeweils nur Teilmengen von Agenten betreffen. Die besten Aktionen der Agenten können dann durch lokale Optimierung bestimmt werden. Entscheidend für den Erfolg dieser Methode ist natürlich, das MAS so zu gestalten, dass globale Abhängigkeiten möglichst vermieden werden und lokale Optimierung global gutes Verhalten erzeugen kann.

3 Multi-Roboter-Systeme

Bei der Anwendung der MAS-Techniken auf Multi-Roboter-Systeme treten zusätzliche Schwierigkeiten auf. Im Gegensatz zu simulierten Agenten muss bei realen Robotern der Zustand des Roboters und der Umgebung aus Sensormessungen geschätzt werden. Für die

Lokalisierung mobiler Roboter bei gleichzeitiger Kartierung der Umgebung können verschiedene SLAM-Verfahren eingesetzt werden [12]. Grundlage hierfür bilden häufig Laser-Scanner, die Abstände in einem weiten Sichtbereich messen, oder Kameras. Objekterkennung und -verfolgung erfolgen häufig kamerabasiert, z.B. mit randomisierten Entscheidungsbäumen (Hough Forests [16]). Wenn 3D-Messungen vorhanden sind, kommen beispielsweise Verfahren zum Einsatz, die Objekte durch Paare von Oberflächenstücken bzw. Randteilen modellieren, detektieren und deren Pose schätzen [8].

Obwohl in den letzten Jahren deutliche Fortschritte bei der Roboterwahrnehmung erzielt wurden, beispielsweise bei der Personenwahrnehmung mittels des Microsoft Kinect-Sensors [54], machen begrenzt leistungsfähige Sensoren sowie vielfältige Variabilitäten und Verdeckungen immer noch Probleme. Um die Wahrnehmungsproblematik zu entschärfen, wird häufig die Umgebung vereinfacht. Beispielsweise wird der Boden frei von Hindernissen gehalten, die vom Roboter nicht gemessen werden können. Objekte werden farblich markiert oder leicht erkennbare Markierungen (z.B. RFID-Tags, QR-Codes) werden an Objekten und Orten angebracht.

Auf Grundlage der Wahrnehmung müssen die Roboter den Weltzustand auch gezielt beeinflussen können. Bei mobilen Robotern steht dabei zunächst die eigene Fortbewegung im Vordergrund. Je nach Umgebung werden dafür verschiedenste Antriebe eingesetzt. Am häufigsten genutzt werden Radroboter, die auf glatten Flächen fahren. Hier sind vor allem Differentialantriebe und Antriebe mit einzeln lenkbaren Rädern üblich. Für unwegsames Gelände und Treppen werden Kettenantriebe mit beweglichen Auslegern eingesetzt. Bei Flugrobotern sind in den letzten Jahren Multi-Rotor-Plattformen populär geworden, die auch auf der Stelle schweben können. Da die Bewegung der Roboter stets mit Unsicherheiten behaftet ist, beispielsweise können Räder durchdrehen oder Flugroboter durch Wind abdriften, ist häufig eine probabilistische Bewegungsmodellierung erforderlich [55].

Für die gezielte Objektmanipulation werden mobile Roboter mit beweglichen Armen und Greifern ausgestattet. Zu dem Navigationsproblem kommen bei der mobilen Manipulation Probleme wie Greifplanung und Redundanzauflösung zwischen Armen und der beweglichen Basis hinzu [27].

Während bei Einzelrobotern höhere Kosten akzeptabel sein können, sind bei Multi-Roboter-Systemen niedrige Kosten pro Roboter besonders wichtig. Daher sind hier die Roboter häufig nur mit einfachen Sensoren, schwachen Aktuatoren, geringer Rechenleistung und beschränkten Kommunikationsmodulen ausgestattet. Beispiele wie der iRobot Roomba-Staubsaugerroboter oder der Parrot AR.Drone-Flugroboter zeigen allerdings, dass durch industrielle Massenfertigung erstaunliche technische Leistungsfähigkeit auch zu einem günstigen Preis realisiert werden kann.

3.1 Roboterschwärme

Roboterschwärme bestehen aus zahlreichen einfachen Robotern, die nur implizit über die Umwelt interagieren [47]. Diese sind durch soziale Insekten, wie Ameisen und Bienen, sowie durch herden- oder schwarmbildende Wirbeltiere (z.B. Fische, Paarhufer, Vögel) inspiriert. Obwohl es offenbar keine zentrale Koordination gibt, sind solche Sozialverbände zu Leistungen in der Lage, die weit über die Fähigkeiten des einzelnen Tiers hinausgehen. Ein Beispiel dafür ist der gemeinsame Transport großer Beutetiere durch mehrere Ameisen. Eigenschaften der natürlichen Vorbilder, wie Robustheit, Flexibilität und Skalierbarkeit sollen durch Roboterschwärme nachvollzogen werden.

Wegen der Schwierigkeit, eine große Zahl von Robotern gleichzeitig zu betreiben, wurden viele Ansätze zunächst nur in der Simulation untersucht. Es gibt aber auch Schwarm-Experimente mit realen Robotern. Beispielsweise entwickelten Beckers et al. [1] einfache Roboter, die unabhängig voneinander in einer Arena fahren, Hindernisse vermeiden und zufällig verteilte Objekte an einer Stelle anhäufen. Dabei folgen die Roboter einfachsten lokalen Regeln, die zu einem globalen Effekt führen.

Turgut et al. [57] untersuchten die Bildung von Roboterschwärmen mit Radrobotern, die Abstand und Richtung ihrer Nachbarn bestimmen können. Durch Angleichung der eigenen Richtung an die Richtung der Nachbarn und Abstandsregulierung zu diesen – nicht zu nah, um Kollisionen zu vermeiden und nicht zu weit weg, um den Schwarm zusammen zu halten – entsteht eine gemeinsame Bewegung.

Ein besonders beeindruckendes Beispiel für kooperatives Verhalten in einem Roboterschwarm ist die Arbeit von Nouyan et al. [37]. Hier löst eine Gruppe identischer Roboter eine Such- und Transportaufgabe. Dazu müssen die Roboter das zu transportierende Objekt finden, einen Pfad zum Zielort bilden, Teamkollegen rekrutieren und gemeinsam das Objekt zum Ziel ziehen. Die Roboter lösen die Aufgabe indem sie einfachen, lokalen Regeln folgen.

3.2 Kommunikations- und Kontrollarchitekturen

Die nur implizite Kommunikation der Roboter mithilfe von Effekten in der Umgebung verhindert eine explizite Koordination der Roboter. Explizite Kommunikation erlaubt es den Robotern sich über den Zustand der Welt auszutauschen, Rollen zu verhandeln und Aktionen zu synchronisieren. Dazu müssen aber Kommunikationsmechanismen und -protokolle etabliert werden, die unzuverlässig und fehleranfällig sein können.

Die Gestaltung der Kommunikations- und Kontrollarchitektur hat daher großen Einfluss auf die Leistungsfähigkeit, Robustheit und Skalierbarkeit eines Mehrrobotersystems. Hierbei können zentralisierte,

hierarchische und verteilte Architekturen unterschieden werden.

Ein zentralisierter Ansatz hat dabei den Vorteil, dass eine zentrale Kontrollinstanz die Übersicht über das gesamte zu lösende Problem hat und so global gute Lösungen finden kann, die einem Roboter mit lokaler Sicht nicht zugänglich sind. Allerdings ist dieser Ansatz mit dem Problem behaftet, dass ein Versagen der Zentrale das Gesamtsystem zerstört. Auch skalieren zentralisierte Methoden schlecht, da die erforderliche Kommunikation mit der Zahl der zu koordinierenden Roboter steigt und insbesondere da die Komplexität der Planungsprobleme explodiert. Ein Beispiel für einen zentralen Koordinierungsansatz ist die Arbeit von Kim et al. [28], die Beobachtungsposen für eine Gruppe von Roboterpaaren optimieren, um möglichst schnell Radioquellen zu lokalisieren.

Die Probleme zentralisierter Ansätze können abgemildert werden, wenn eine Hierarchie zwischen den Kontrollinstanzen etabliert wird. Jede Kontrollinstanz koordiniert nur eine kleine Zahl von untergeordneten Instanzen bis letztendlich die einzelnen Roboter auf unterster Ebene mit der Umwelt interagieren. Diese Vorgehensweise führt dazu, dass in jeder Kontrollinstanz handhabbare Planungsprobleme entstehen. Andererseits wird durch die Kontrollhierarchie die Koordination von Robotern erschwert, die weit im Hierarchiebaum voneinander entfernt sind. Auch ist das Versagen von hoch in der Hierarchie stehenden Instanzen weiterhin problematisch. Beispiele für hierarchische Kontrollarchitekturen sind die von Haghig und Cheah vorgeschlagene Multi-Gruppen-Koordination [20], mit der große Roboterschwärme in komplexe Formationen gebracht werden können, und die hierarchisch-reaktive Architektur [2, 3], die sich beim Roboterfußball bewährt hat.

Die meisten Kontrollarchitekturen verfolgen einen verteilten Ansatz, beispielsweise die von Parker vorgeschlagene Alliance-Architektur [39]. Hier hängen die Aktionen der Roboter nur von ihrer lokalen Wahrnehmung und der Kommunikation mit benachbarten Robotern ab. Diese Vorgehensweise ist sehr robust, da jeder Teil ausfallen kann ohne das Gesamtsystem zu zerstören, und skaliert gut, da es keine Engstellen in Kommunikation oder Kontrolle gibt.

Es besteht aber die Schwierigkeit, in verteilten Architekturen global konsistentes Verhalten zu erzeugen. Um auch bei wechselnder Topologie des Kommunikationsgraphen, variablen Kommunikationslatenzen und Unterbrechungen einen Konsens zwischen mehreren Agenten herbeizuführen wurden verschiedene Algorithmen in der Literatur vorgeschlagen bzw. Konvergenzkriterien untersucht, z.B. von Olfati-Saber und Murray [38], Moreau [35], Xiao und Wang [62] sowie Lorenz und Lorenz [33]. Das generelle Prinzip ist dabei die gewichtete Mittelung des Zustands benachbarter Agenten.

Ein weiteres Problem dezentraler Ansätze ist deren

formale Beschreibung und Verifikation. Klavins und Murray [29] haben für diesen Zweck die Computation and Control-Sprache CCL entwickelt, die es ermöglicht gewünschte Eigenschaften kooperativer Robotersysteme zu beweisen.

Hybride Architekturen kombinieren lokale Kontrolle mit zentralisierten Ansätzen, um beispielsweise globale Ziele für ein Roboterteam vorzugeben oder dessen Arbeit zentral zu überwachen. Ein Beispiel ist die von Michael et al. vorgeschlagene Asymmetric-Broadcast-Kontrollarchitektur [34], in der eine Kontrollinstanz auf einem abstrakten Zustand agiert, dessen geringe Dimensionalität unabhängig von der Zahl der Roboter ist. Dieser kann beispielsweise aus den Zuständen der Einzelroboter aggregiert werden (z.B. Schwerpunkt und weitere Momente der Roboterpositionen) oder aus der zu lösenden Aufgabe hergeleitet sein (z.B. Pose des zu transportierenden Objekts). Die Kontrollinstanz beeinflusst die Roboter durch Broadcast-Nachrichten, muss also nicht jeden einzelnen Roboter adressieren. Michael et al. demonstrieren mit diesem Ansatz die Bewegung von Formationen von Bodenrobotern, die durch einen Flugroboter überwacht werden, und den kooperativen Transport eines großen Objekts, das durch mehrere Roboter umschlossen wird.

3.3 Aufgabenzuordnung

Wenn Aufgaben in unabhängige Teilaufgaben zerlegt werden können und Roboter selbständig in der Lage sind, diese zu lösen, kann die Kooperation auf der Ebene der Aufgabenzuordnung organisiert werden. Für die Gesamteffizienz ist dabei die Zuordnung von Teilaufgaben zu Robotern entscheidend, wobei Problemstellungen mit Spezialrobotern, die nur eine Teilaufgabe bearbeiten können, und solche mit Mehrzweckrobotern, die mehrere Teilaufgaben beherrschen, unterschieden werden [18].

Bei der Aufgabenzuordnung muss beachtet werden, ob für die Erledigung einer Teilaufgabe ein Roboter ausreichend ist, oder ob mehrere Roboter erforderlich sind. Ein weiterer Aspekt ist, ob die Aufgabenzuordnung instantan immer dann erfolgt wenn eine neue Aufgabe anfällt, oder ob die Zuordnung über einen längeren Zeitraum optimiert werden kann.

Im Allgemeinen ist die Aufgabenzuordnung ein kombinatorisches Optimierungsproblem, bei dem für jeden Roboter ein Schätzwert verwendet wird, welche Kosten bei der Erledigung einer bestimmten Aufgabe anfallen werden und welcher Nutzen zu erwarten ist.

Der einfachste Fall der Aufgabenzuordnung für einzeln lösbare Aufgaben, Spezialroboter, und instantane Zuordnung ist ein Beispiel für ein optimales Zuordnungsproblem, das bei Vorhandensein einer zentralen Kontrollinstanz effizient durch Verfahren der linearen Programmierung gelöst werden kann [30]. In verteilten Systemen kann die Aufgabenzuordnung hier durch Auktions-

verfahren [17] oder eine verteilte Version des Simplex-Algorithmus [6] realisiert werden.

Für die Aufteilung eines Arbeitsraums in konvexe Gebiete, deren Anfragen von jeweils einem Agenten bedient werden, sodass sich die Anfragen gleichmäßig auf alle Agenten verteilen, haben Pavone et al. [42] einen verteilten Algorithmus vorgeschlagen, der mit dem Lloyd-Algorithmus zur Berechnung von Voronoi-Diagrammen verwandt ist.

Für die Zuordnung mobiler Agenten mit begrenztem Kommunikationsradius zu verteilten Zielpunkten haben Smith und Bullo [49] monotone Algorithmen vorgeschlagen – die Anzahl der belegten Zielpunkte erhöht sich monoton – die höchstens einen konstanten Faktor schlechter sind als optimale monotone Algorithmen.

Die komplexeren Zuordnungsproblem-Varianten sind in der Regel NP-hart und daher nur für kleine Problemgrößen optimal lösbar. Hier müssen approximative Lösungen akzeptiert werden.

Wenn die Teilaufgaben in einer bestimmten Reihenfolge oder synchronisiert bearbeitet werden müssen, entstehen zusätzliche Abhängigkeiten, die das Zuordnungsproblem weiter erschweren. Einen Überblick über exakte und heuristische Verfahren zur Lösung dieser komplexen Zuordnungsprobleme geben Brucker und Knust [5].

3.4 Pfadplanung

Ein zentrales Problem bei mobilen Multi-Roboter-Systemen ist die Pfadplanung, d.h. die Frage wie die Roboter in einem gemeinsamen Arbeitsraum von einer Startkonfiguration in eine Zielkonfiguration bewegt werden können, ohne dass sie sich gegenseitig stören oder gar kollidieren, bzw. wie sie sich als Gruppe gemeinsam bewegen können, um bestimmte Bewegungsziele zu erreichen [40]. Dabei ist insbesondere auch die Berücksichtigung bewegter Hindernisse wichtig. Weil zentralisierte Pfadplanung für mehrere Roboter nicht skaliert, wurden Methoden entwickelt, die das Planungsproblem in handhabbare Teilprobleme zerlegen und so eine Balance zwischen Optimalität und Effizienz finden.

Bei der Pfadplanung kann man Offline- und Online-Probleme unterscheiden. Während bei der Offline-Planung die Bewegungen aller Roboter vorgeplant werden, entsteht der Pfad bei der Online-Planung zur Ausführungszeit, was die Berücksichtigung von Veränderungen in einer dynamischen Umwelt und die Zuweisung neuer Aufgaben erlaubt. Neben der Kollisionsvermeidung werden bei der Pfadplanung oft weitere Kriterien optimiert, wie die Pfadlänge, die benötigte Zeit oder der Energieverbrauch. Auch die Nähe zu Hindernissen kann mit Kosten belegt werden. Dabei müssen Nebenbedingungen eingehalten werden, welche die Navigationsfähigkeiten der Roboter beschreiben (maximale Geschwindigkeit, maximaler Anstieg, mögliche Bewegungsrichtungen) oder Bedingungen für die gemeinsame

Bewegung einer Gruppe von Robotern formalisieren.

Pfadplanungsverfahren können nach verschiedenen Kriterien beurteilt werden. Ein Kriterium ist die Vollständigkeit, d.h. ob immer ein Pfad gefunden wird wenn eine Lösung existiert. Auch Komplexität und Optimalität der Planung sind wichtige Kriterien. Ein anderer Aspekt ist die für die Planung benötigte Information. Verfahren, die globale Information benutzen und direkt im Zustandsraum aller Roboter planen, werden zentralisiert genannt. Hier können klassische vollständige Planungsverfahren, wie A^* [22], optimale Lösungen finden – allerdings für den Preis einer Planungskomplexität, die exponentiell mit der Zahl der Roboter steigt. In hochdimensionalen Konfigurationsräumen kommen häufig Sampling-basierte Verfahren, wie RRTs [31] zum Einsatz, die eine explizite Hindernisrepräsentation vermeiden indem sie zufällige Pfadstücke auf Kollisionsfreiheit testen und bei erfolgreichem Test zum Routengraphen hinzufügen.

Besonders einfache Verfahren zur Hindernisvermeidung sind Potentialfeld-Methoden, die dem Gradienten eines Potentialfelds folgen, das die Roboter von Hindernissen abstößt und zum Ziel hin zieht (z.B. [56, 53, 9]). Hier besteht allerdings die Gefahr des Steckenbleibens in lokalen Minima, wie Sackgassen. Eine weitere Möglichkeit der Komplexitätsreduktion ist, in der Umgebung ein Routennetz – wie einen Straßengraphen oder Flugkorridore – vorzuberechnen, auf denen sich die Roboter bewegen müssen. Ein Beispiel für einen Routennetz-basierten Ansatz ist die Arbeit von Svestka und Overmars [52], die zunächst für einzelne Roboter Routengraphen erzeugen, diese zu einem kombinierten Routengraphen, der als Knoten die Position aller Roboter enthält, zusammenfassen und dort kollisionsfreie Pfade finden.

Bei einer großen Roboterzahl ist es notwendig, die Planungsprobleme zu entkoppeln und nach lokal guten Lösungen zu suchen – unter Aufgabe von Vollständigkeit und globaler Optimalität. Bei entkoppelten Ansätzen können grob zwei Vorgehensweisen unterschieden werden: priorisierte Planung und Pfadkoordination. Bei der priorisierten Planung [14] werden die Roboter einer nach dem anderen betrachtet. Bereits geplante Roboter werden als bewegte Hindernisse berücksichtigt, wozu der Konfigurationsraum um die Zeit-Dimension erweitert werden muss. Die Reihenfolge der Roboter entspricht dabei der Priorität. Beispielsweise können Roboter zuerst geplant werden, die besonders dringende Aufgaben zu erledigen haben oder es kann eine Reihenfolge gesucht werden, welche die Gesamtlänge der Pfade minimiert [4].

Bei der Pfadkoordination werden zunächst Pfade für alle Roboter unabhängig voneinander bestimmt und dann wird für jeden Roboter ein Geschwindigkeitsprofil gefunden [26]. Dabei müssen Roboter an geeigneten Stellen warten bis blockierte Pfadteile durch andere Roboter wieder freigegeben werden. Allerdings kann es dabei zu Deadlock-Situationen kommen, in denen mehrere Roboter gegenseitig aufeinander warten. Diese Situationen

müssen getrennt detektiert und z.B. durch einen lokal begrenzten zentralisierten Planer behandelt werden.

Eng verwandt mit dem Pfadplanungsproblem ist das Problem der Bewegungskoordination mehrerer Roboter, die jeweils eigene Start- und Zielpositionen haben. Hier wird eine roboterzentrische Perspektive eingenommen und die Aufgabe ist, einen Pfad abzuarbeiten. Dabei müssen auch unerwartet auftauchende Hindernisse vermieden und eventuell auftretende Konflikte mit anderen Robotern aufgelöst werden. Dies kann beispielsweise durch lokale Pfadplanung, auf der Grundlage der aktuellen Hinderniswahrnehmung, oder durch Beachtung von Vorfahrtsregeln geschehen. Da die Roboter in der Regel nicht instantan stoppen können, ist dabei die Berücksichtigung der eigenen Geschwindigkeit und der Vorhersage der Bewegung der Roboter in der Umgebung wichtig [36].

Ein weiteres verwandtes Problem ist, die Konnektivität in einem Netzwerk sich bewegendes Roboter sicher zu stellen. Die Arbeit von Zavlanos et al. [63] bietet hierzu einen Überblick.

3.5 Lernen

Aufgrund der komplexen Wechselwirkungen mit der Umgebung und untereinander ist es oft nicht praktikabel, das Verhalten von Robotern in Multi-Roboter-Systemen auszuprogrammieren oder automatisch von Modellen abzuleiten. Auch können sich Umgebung oder Roboter mit der Zeit verändern, sodass das Roboterverhalten angepasst werden muss. In solchen Situationen können Verfahren des optimierenden Lernens (Reinforcement Learning, RL [51]) zum Einsatz kommen. Hier interagiert ein Roboter mit einer Umgebung und erhält Feedback durch Belohnungen oder Bestrafungen. Diese Information ist weniger aussagekräftig als im überwachten Lernen, wo für jede Situation die gewünschte Aktion vorgegeben wird. Da Aktionen sich nicht immer sofort in Belohnungen oder Bestrafungen niederschlagen, sondern häufig erst nach einer Reihe anderer Aktionen klar wird, ob der Agent erfolgreich war oder nicht, müssen die Belohnungen oder Bestrafungen zeitlich zurückpropagiert werden.

Sowohl für den Fall, dass dem Agenten ein Modell der Welt zur Verfügung steht, als auch für den Fall dass der Agent die Auswirkungen von Aktionen auf den Weltzustand mitlernen muss, gibt es RL-Verfahren, die bereits vielfältig für einzelne Roboter eingesetzt werden. Die Anwendung von RL auf Multi-Roboter-Systeme ist deutlich anspruchsvoller [7]. Häufig ist nicht a priori klar, wie die Belohnungen oder Bestrafungen gewählt werden müssen, um ein gewünschtes Systemverhalten zu erzielen. Auch muss jeder Lerner die Verhaltensänderungen der anderen lernenden Roboter berücksichtigen. Nur so kann eine Koordination mehrerer Roboter erreicht werden, beispielsweise die Wahl einer gemeinsamen Ausweichrichtung wenn ein Roboter-

paar zusammen bleiben muss. Diese Nichtstationarität verletzt die Annahme einer stationären Umgebung, die für den Konvergenzbeweis klassischer RL-Verfahren notwendig ist.

Ein weiteres Problem bei der Anwendung von RL-Verfahren auf Multi-Roboter-Systeme ist die Dimensionalität des Lernproblems. Während schon bei einzelnen Robotern häufig nicht genug Erfahrungen gemacht werden können, um alle Parameter der Lernmaschine hinreichend zu bestimmen, tritt diese Problematik bei Multi-Roboter-RL noch stärker zutage.

Andererseits gibt es aber auch Vorteile durch den Einsatz mehrerer Roboter im RL. Beispielsweise können Roboter, die sich genügend ähnlich sind, Erfahrungen austauschen und so schneller als ausschließlich aus eigener Erfahrung lernen. Eine besonders wirkungsvolle Methode der Nutzung fremder Erfahrungen ist die Imitation eines Vorbilds, das eine Aufgabe bereits lösen kann, wenn der lernende Roboter in einer Situation über zu wenig eigene Erfahrungen verfügt. Beispielsweise kann ein von einem Roboter als befahrbar klassifizierter Weg von einem anderen Roboter wieder verwendet werden.

Ein Beispiel für die erfolgreiche Anwendung von RL-Verfahren in Multi-Roboter-Systemen ist das Fangen eines „Beute“-Roboters durch mehrere „Jäger“-Roboter [25] welche lernen die Beute zu umzingeln und in die Enge zu treiben. Auch im Roboterfußball wurden RL-Techniken erfolgreich eingesetzt, z.B. von Riedmiller et al. [46] für das Dribbeln des Balls mit omnidirektionalen Radrobotern und für Verteidigungs- und Angriffsverhalten in der Simulationsliga.

4 Anwendungsbeispiele

Auch wenn viele Fragen bei Multi-Roboter-Systemen noch Gegenstand der Forschung sind, gibt es doch erste Anwendungen, beispielsweise im Versandhandel, bei der Konstruktion von Strukturen, der Exploration unbekannter Umgebungen oder beim Teamspiel.

4.1 Hunderte Transportroboter im Logistikzentrum

Eines der beeindruckenden Beispiele für kooperierende mobile Roboter wurde von der Firma Kiva Systems für den Bereich der Lager-Logistik im Versandhandel realisiert [61]. Für diese Anwendung hat Kiva autonome Transportroboter entwickelt, die in der Lage sind, unter Regale zu fahren, diese hochzuheben und mit dem Regal durch große Hallen zu navigieren (Abb. 2).

In den Regalen befinden sich Lagerplätze für unterschiedlichste Waren, wie beispielsweise Bücher. Die Roboter orientieren sich mittels Odometrie und am Boden angebrachter Matrixcodes. Soll nun ein bestellter Artikel in einen Versandkarton gepackt werden, bringt ein Roboter das passende Regal in der richtigen Ausrichtung



Bild 2: Kiva-Transportroboter im Versandhandel [61].

aus dem Lager zu einer Packstation. Das Herausnehmen des Artikels übernimmt ein Mensch, dem durch ein Lichtsignal der Lagerplatz im Regal angezeigt wird. Anschließend bringt der Roboter das Regal wieder zurück auf einen freien Stellplatz in der Halle. Auch der Abtransport fertig gepackter Versandkartons und die Wiederauffüllung der Regale geschehen mit Hilfe der Roboter.

Durch den Einsatz hunderter Roboter, die sich an den Packstationen anstellen, wird erreicht, dass alle sechs Sekunden ein neuer Artikel gepackt werden kann. Dies ist eine deutliche Erhöhung der Produktivität gegenüber bisherigen Lösungen. Neue Aufträge werden ständig ins System eingespeist und können in Minuten gepackt werden. Ein weiterer Vorteil der Kiva-Lösung ist die Redundanz durch den Einsatz vieler identischer Roboter.

Die Kiva-Lösung ist ein Multi-Agenten-System, in dem jede Komponente nur Zugriff auf die lokal verfügbare Information hat. Jeder Roboter und jede Packstation wird durch einen Agenten repräsentiert. Ein Job-Manager ist für die systemweite Ressourcenzuteilung zuständig. Die Agenten kommunizieren über XML-Nachrichten, von denen es über einhundert verschiedene Typen gibt.

Die Transportroboter planen selbständig Pfade mit dem A*-Algorithmus und entscheiden mit Hilfe von Prioritäten auch, wie Konflikte aufgelöst werden, beispielsweise wenn ein Regal gleichzeitig von mehreren Packstationen angefordert wird. Ressourcenzuordnungsprobleme werden nicht optimal gelöst, sondern durch kostenbasierte Heuristiken entschieden. Beispielsweise werden einer Packstation möglichst ähnliche Aufträge zugeordnet, um die Wahrscheinlichkeit zu erhöhen, dass ein Regal für mehrere Bestellungen benötigt wird. Auch werden die Stellplätze der Regale so optimiert, dass häufig benötigte Artikel in der Nähe der Packstationen stehen.

Seitdem 2005 die erste Pilotanlage in Betrieb ging, hat Kiva eine große Zahl von Versandzentren automatisiert. Im Jahr 2012 wurde Kiva Systems für einen signifikanten Betrag von Amazon erworben, was die Nützlichkeit des realisierten Ansatzes unterstreicht.

4.2 Kooperative Konstruktion durch Flugroboter

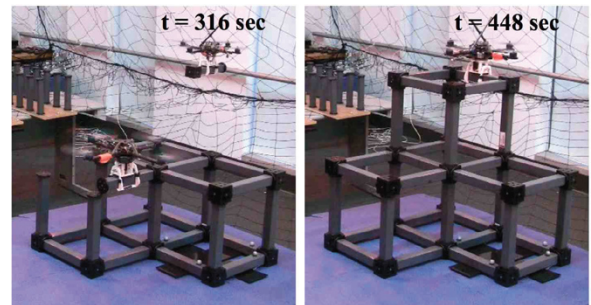


Bild 3: Konstruktion von Strukturen mit Flugrobotern [32].

Ein weiteres beeindruckendes Beispiel für die Kooperation mobiler Roboter ist die Arbeit von Lindsey et al. [32]. Sie benutzen Quadcopter, um gerüstartige Strukturen aus zwei Arten von Einzelteilen – Verbinder und Quadratrohre – zusammenzusetzen. Die Enden der Teile enthalten Magnete, sodass keine Schraubverbindungen nötig sind. Aufgrund der Montage von oben dürfen die Strukturen keine Überhänge oder Löcher haben.

Die Strukturen werden schichtweise aufgebaut, wobei jede Schicht zeilenweise abgearbeitet wird. Die Flugroboter holen die Teile aus einer geordneten Ablage ab. Dazu sind sie mit einem Greifer ausgestattet. Die Pose der Roboter wird durch ein Motion-Capture-System ermittelt, das den gesamten Arbeitsraum abdeckt.

Für die Koordination der Flugroboter wird ein Zustandsautomat benutzt, der durch Warteschlangen sicherstellt, dass jeweils nur ein Roboter ein Teil aufnimmt oder in die Struktur einbaut. Die Wartezonen sind so angeordnet, dass die Roboter sich in einer Schleife bewegen und so Kollisionen vermieden werden.

Der Ansatz wurde in Simulation und mit bis zu drei realen Robotern für die Konstruktion von Strukturen mit bis zu 40 Teilen getestet (Abb. 3). Während zwei Roboter die Konstruktionszeit halbieren, bringt der Einsatz eines dritten Roboters kaum noch Gewinn, da die Roboter dann häufig warten müssen. Durch gleichzeitiges Einfügen von Teilen an mehreren Stellen und mehrere Teilelager könnten größere Roboterteams größere Strukturen in kurzer Zeit errichten. Um stabilere Strukturen zu erreichen, müsste eine belastbarere Verbindungstechnik eingesetzt werden als die derzeit genutzten magnetischen Schnappverschlüsse.

4.3 Multi-Roboter-Exploration

Eine wichtige Aufgabe für Roboter ist die Exploration unbekannter Gebiete. Durch den gleichzeitigen Einsatz mehrerer Roboter können die Ausfallsicherheit des Systems erhöht und die Explorationszeit verkürzt werden. Auch kann durch die Fusion von Beobachtungen aus



Bild 4: Erkundung einer Büroumgebung [60].

verschiedenen Blickpunkten die Beobachtungssicherheit erhöht werden. Es besteht aber auch die Gefahr, dass sich die Roboter gegenseitig stören. Daher ist es wichtig, für mehrere Roboter einen guten Explorationsplan zu entwickeln. In der Arbeit von Wurm et al. [60] wird das Ziel verfolgt, die Explorationszeit zu minimieren. Dazu wird mit Hilfe der bis zu einem Zeitpunkt erstellten Teilkarte die Umgebung in Segmente eingeteilt. Anhand der Fahrtkosten werden noch nicht vollständig explorierte Segmente, in denen bekannte Gebiete an unbekannte Gebiete grenzen, auf die Roboter aufgeteilt. Die Segmente werden dabei durch Voronoi-Zerlegung des Freiraums erzeugt und korrespondieren in den getesteten Büroumgebungen häufig mit einzelnen Räumen.

Der Ansatz wurde in Simulation mit bis zu acht Robotern und mit zwei realen Robotern getestet (Abb. 4), wobei sich eine interessante Strategie ergeben hat. Während ein Roboter zunächst den Korridor der Büroumgebung vollständig kartiert, exploriert der andere Roboter schon die einzelnen Räume. Insgesamt konnte so eine deutliche Reduktion der Explorationszeit erreicht werden. Ein solcher Ansatz setzt natürlich voraus, dass die Roboter ihre Messungen stets in einer gemeinsam genutzten Karte fusionieren. Im Falle von Kommunikationseinschränkungen kann diese Voraussetzung durch die Planung von Kommunikations-Rendezvous relaxiert werden.

Ein weiteres Beispiel für Multi-Roboter-Exploration ist die Arbeit von Haumann et al. [24, 23], welche die Roboterorientierung bei der Optimierung der nächsten Zielpunkte mit berücksichtigt und so die Exploration beschleunigt.

4.4 Roboterfußball

Während bei den bisherigen Anwendungsbeispielen alle Roboter ein gemeinsames Ziel verfolgen, treffen bei Teamsportarten zwei Gruppen von Robotern aufeinander, die gegensätzliche Ziele verfolgen. Daher besteht die Umgebung nicht aus statischen Elementen oder leicht vorhersagbaren Agenten, sondern die Aktionen der generischen Spieler sind nur bedingt voraussehbar. Ein besonders prominentes Beispiel für Roboterkooperation ist

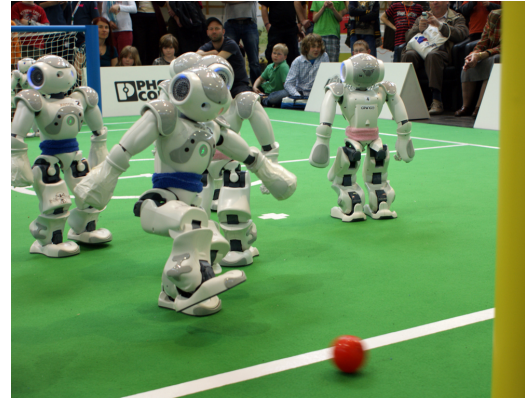


Bild 5: Fußball in der RoboCup Standard-Plattform-Liga.

der Roboterfußball. Hier werden seit 1997 internationale Wettbewerbe von der RoboCup Federation veranstaltet, mit dem langfristigen Ziel bis 2050 den FIFA-Weltmeister schlagen zu können.

Natürlich sind die Roboter von diesem Ziel noch weit entfernt, aber es gibt jedes Jahr Fortschritte zu beobachten. Die Wettbewerbe werden in verschiedenen Ligen ausgetragen, in denen unterschiedliche Teilaspekte des Fußballproblems untersucht werden. In der 2D-Simulationsliga spielen stark vereinfachte Roboter. Hier wird vor allem Teamspiel und Taktik untersucht. Es gibt auch eine 3D-Simulationsliga, in der komplexe humanoide Roboter antreten. Hier, wie in den Ligen mit realen zweibeinigen Robotern, stehen vor allem die robuste und zügige Fortbewegung auf zwei Beinen und die Erzeugung dynamischer Ganzkörperbewegungen im Vordergrund.

Es gibt auch zwei Ligen für Radroboter, in denen Teamspiel schon recht gut ausgeprägt ist. In der SmallSize-Liga sind die Roboter farbig markiert und werden durch Kameras über dem Spielfeld beobachtet, was das Wahrnehmungsproblem enorm vereinfacht. Die Steuerung eines ganzen Teams erfolgt durch einen externen Computer, sodass der Einsatz zentraler Koordinierungsstrategien möglich wird. Das Spiel der SmallSize-Roboter ist inzwischen so schnell, dass menschliche Beobachter kaum noch folgen können, und beinhaltet fortgeschrittene Elemente wie Hochschüsse und Doppelpässe. Auch die Roboter der MidSize-Liga spielen sehr dynamisch und müssen den Ball bei einem Angriff mindestens einmal abgeben.

Für die humanoiden Fußballroboter gibt es die Standard-Plattform-Liga, in der alle Teams den Nao-Roboter der französischen Firma Aldebaran verwenden (Abb. 5). Hier spielen vier Roboter pro Team, die sich beispielsweise über die Ballposition austauschen und auch abstimmen, wer zum Ball geht und wer andere Aufgaben übernimmt. Wie im menschlichen Fußball werden Rollen häufig mit Hilfe von Regionen vergeben, für die einzelne Roboter zuständig sind. Die Zuordnung ist dabei nicht statisch, sondern wird entsprechend der Spiel-

situation angepasst.

Auch in der Humanoid-Liga gibt es ähnliche Mechanismen für Teamspiel, die allerdings weniger ausgeprägt sind, da in der KidSize-Klasse (bis 60 cm Robotergröße) nur drei Spieler pro Mannschaft und in der TeenSize-Klasse (90-120 cm) nur zwei Spieler pro Mannschaft erlaubt sind. Die größten humanoiden Fußballroboter der AdultSize-Klasse (>130 cm) sind noch nicht robust genug gegen Stürze und wetteifern daher, und weil viele Forschergruppen nur einen großen Roboter haben, noch immer im Elfmeterschuss-Wettbewerb.

In dem Maße, wie die größeren humanoiden Fußballroboter die elementaren Techniken besser beherrschen und preiswerter werden, wird es möglich sein, die Zahl der Spieler zu erhöhen und dann wird die Kooperation innerhalb der Mannschaft entsprechend wichtiger werden. Ein Schritt in diese Richtung ist der vom Team der Universität Bonn kürzlich vorgestellte quelloffene NimbRO-OP-Roboter [48], der anderen Teams den Einstieg in die TeenSize-Klasse erleichtern soll.

5 Diskussion

Kooperierende mobile Roboter sind ein faszinierender Forschungsgegenstand, der deutlich über die Problemstellungen der mobilen Robotik und der Multi-Agenten-Systeme hinausgeht. Um Aufgaben mit Teams mobiler Robotern zu lösen, ist nicht nur eine Synthese der in beiden Bereichen entwickelten Methoden nötig, sondern es stellen sich ganz eigene Herausforderungen, wie die Kostengünstigkeit und Pflegeleichtheit der verwendeten Roboter oder die Zuverlässigkeit der Kommunikation. Dennoch werden kooperierende mobile Roboter nicht nur in Forschungslaboren untersucht, sondern es gibt bereits erste kommerzielle Anwendungen, wie beispielsweise Hol- und Bringedienste in Krankenhäusern, Containerlogistik in Häfen und die Automatisierung von Warenlagern. Nicht zuletzt das Militär interessiert sich für Roboterteams, beispielsweise für Überwachungs- und Transportaufgaben.

Kooperierende mobile Roboter stehen in enger Beziehung zu Sensor-Aktuator-Netzwerken. Durch die Einbettung in intelligente Umgebungen müssen nicht mehr alle Funktionen auf den Robotern selbst realisiert werden. Beispielsweise können die Wahrnehmung des Umgebungszustands und die Posenschätzung der Roboter durch ein in den Raumecken installiertes Mehrkammersystem erfolgen. Auch bei der Beeinflussung der Umgebung können fest installierte Aktuatoren, wie elektrische Türöffner, enorm hilfreich sein. Die Integration von Robotern und intelligenten Umgebungen erfordert allerdings neuartige Middleware-Konzepte [45].

Neben der Kooperation der Roboter untereinander bestehen natürlich auch enge Beziehungen zum Gebiet der Mensch-Roboter-Kooperation. Wenn Menschen und Roboter gemeinsam Aufgaben lösen, kann jede Seite ihre

spezifischen Stärken einbringen. Während die menschliche Intelligenz noch lange nicht von technischen Systemen erreicht werden wird, können die Roboter Teilaufgaben übernehmen, die Stärke, Präzision oder Ausdauer erfordern oder besonders gefährlich sind. Die Kooperation mit menschlichen Teammitgliedern stellt natürlich besondere Herausforderung an die Mensch-Roboter-Interaktion [15].

Aufgrund der Kosten und der technischen Komplexität werden sich kooperierende mobile Roboter zunächst in der industriellen Produktion und im professionellen Dienstleistungsbereich verbreiten sowie vom Militär verwendet werden. Damit dabei nicht immer das Rad neu erfunden werden muss, ist es nötig, Standardlösungen für wiederkehrende Teilprobleme zu entwickeln und – möglichst quelloffen – anderen zur Verfügung zu stellen. Die Middleware ROS [44] ist ein gutes Beispiel für die Dynamik, die durch Open-Source-Software entstehen kann. Um den Einsatz in Umgebungen in denen sich Menschen befinden zu ermöglichen, muss der Aspekt der Sicherheit des Roboterbetriebs stärker adressiert werden. Dies ist ein extrem schwieriges Thema, da vielfältige Gefahren denkbar sind, die zuverlässig erkannt und vermieden werden müssen. Andererseits sind Einwirkungen auf die Umgebung notwendig, um die gewünschten Aufgaben zu erfüllen.

Mit der zunehmend kostengünstigen Massenfertigung von Robotern für Privatanwender und der Verbreitung intelligenter vernetzter Umgebungen sind Szenarien vorstellbar, in denen Roboterteams sich in unseren Alltagsumgebungen nützlich machen. Wer hätte nicht gern zuhause ein paar Einzelroboter, die lästige Haushaltsaufgaben übernehmen, und viele ältere, kranke oder behinderte Menschen sind besonders auf Hilfe angewiesen. Bis diese Visionen Realität werden, muss allerdings noch viel Forschungs- und Entwicklungsarbeit geleistet werden.

Literatur

- [1] R. Beckers, OE Holland, and J.L. Deneubourg. From local actions to global tasks: Stigmergy and collective robotics. In *Artificial life IV*, volume 181, page 189, 1994.
- [2] S. Behnke and R. Rojas. A hierarchy of reactive behaviors handles complexity. In *Balancing reactivity and social deliberation in multi-agent systems*, pages 125–136. Springer, 2001.
- [3] S. Behnke and J. Stückler. Hierarchical reactive control for humanoid soccer robots. *International Journal of Humanoid Robots (IJHR)*, 5(3):375–396, 2008.
- [4] M. Bennet, W. Burgers, and S. Thrun. Finding and optimizing solvable priority schemes for decoupled path planning techniques for teams of mobile robots. *Robotics and Autonomous Systems*, 41(2):89–99, 2002.
- [5] P. Brucker and S. Knust. *Complex scheduling*. Springer, 2011.
- [6] M. Bürger, G. Notarstefano, F. Bullo, and F. Allgöwer. A distributed simplex algorithm for degenerate linear programs and multi-agent assignments. *Automatica*, 48(9):2298–2304, 2012.

- [7] L. Busoniu, R. Babuska, and B. De Schutter. A comprehensive survey of multiagent reinforcement learning. *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on*, 38(2):156–172, 2008.
- [8] C. Choi, Y. Taguchi, L. Tuzel, O. Ming-Yu, and S. Ramalingam. Voting-based pose estimation for robotic assembly using a 3D sensor. In *Robotics and Automation (ICRA), IEEE International Conference on*, pages 1724–1731, 2012.
- [9] D.V. Dimarogonas. Sufficient conditions for decentralized potential functions based controllers using canonical vector fields. *Automatic Control, IEEE Transactions on*, 57(10):2621–2626, 2012.
- [10] M. Dorigo, M. Birattari, and T. Stutzle. Ant colony optimization. *Computational Intelligence Magazine, IEEE*, 1(4):28–39, 2006.
- [11] M. Dorigo, D. Floreano, L.M. Gambardella, et al. Swarmantoid: A novel concept for the study of heterogeneous robotic swarms. *IEEE Robotics & Automation Magazine*, 2012.
- [12] H. Durrant-Whyte and T. Bailey. Simultaneous localization and mapping. *Robotics & Automation Magazine, IEEE*, 13(2):99–110, 2006.
- [13] A.P. Engelbrecht. *Fundamentals of computational swarm intelligence*. Wiley London, 2005.
- [14] M. Erdmann and T. Lozano-Perez. On multiple moving objects. *Algorithmica*, 2:477–521, 1987.
- [15] T. Fong, I. Nourbakhsh, and K. Dautenhahn. A survey of socially interactive robots. *Robotics and autonomous systems*, 42(3):143–166, 2003.
- [16] J. Gall, A. Yao, N. Razavi, L. Van Gool, and V. Lempitsky. Hough forests for object detection, tracking, and action recognition. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 33(11):2188–2202, 2011.
- [17] B.P. Gerkey and M.J. Mataric. A market-based formulation of sensor-actuator network coordination. In *Proceedings of the AAAI Spring Symposium on Intelligent Embedded and Distributed Systems*, 2002.
- [18] B.P. Gerkey and M.J. Mataric. A formal analysis and taxonomy of task allocation in multi-robot systems. *Robotics Research, International Journal of*, 23(9):939–954, 2004.
- [19] C. Guestrin, D. Koller, and R. Parr. Multiagent planning with factored MDPs. *Advances in neural information processing systems*, 14:1523–1530, 2001.
- [20] R. Haghighi and C.C. Cheah. Multi-group coordination control for robot swarms. *Automatica*, 48(10):2526–2534, 2012.
- [21] J.C. Harsanyi and R. Selten. *A general theory of equilibrium selection in games*. The MIT Press, 2003.
- [22] E.E. Hart, N.J. Nilsson, and B. Raphael. A formal basis for the heuristic determination of minimum cost paths. *System Science and Cybernetics, IEEE Transactions on*, 4(2):100–107, 1968.
- [23] A. D. Haumann, A. Breitenmoser, V. Willert, K. D. Listmann, and R. Siegwart. DisCoverage for non-convex environments with arbitrary obstacles. In *Robotics and Automation (ICRA), IEEE International Conference on*, pages 4486–4491, 2011.
- [24] A. D. Haumann, K. D. Listmann, and V. Willert. DisCoverage: A new paradigm for multi-robot exploration. In *Robotics and Automation (ICRA), IEEE International Conference on*, pages 929–934, 2010.
- [25] Y. Ishiwaka, T. Sato, and Y. Kakazu. An approach to the pursuit problem on a heterogeneous multiagent system using reinforcement learning. *Robotics and Autonomous Systems*, 43(4):245–256, 2003.
- [26] K. Kant and S.W. Zucker. Toward efficient trajectory planning: The path-velocity decomposition. *Robotics Research, The International Journal*, 5(3):72–89, 1986.
- [27] O. Khatib. Mobile manipulation: The robotic assistant. *Robotics and Autonomous Systems*, 26(2):175–183, 1999.
- [28] C.Y. Kim, D. Song, Y. Xu, and J. Yi. Localization of multiple unknown transient radio sources using multiple paired mobile robots with limited sensing ranges. In *Robotics and Automation (ICRA), IEEE International Conference on*, pages 5167–5172, 2011.
- [29] E. Klavins and R.M. Murray. Distributed algorithms for cooperative control. *Pervasive Computing, IEEE*, 3(1):56–65, 2004.
- [30] H. W. Kuhn. The hungarian method for the assignment problem. *Naval Research Logistics Quarterly*, 2, 1955.
- [31] S. M. LaValle and J. J. Kuffner. Randomized kinodynamic planning. *International Journal of Robotics Research*, 20(5):378–400, 2002.
- [32] Q. Lindsey, D. Mellinger, and V. Kumar. Construction of cubic structures with quadrotor teams. *Proc. Robotics: Science & Systems VII*, 2011.
- [33] J. Lorenz and D.A. Lorenz. On conditions for convergence to consensus. *Automatic Control, IEEE Transactions on*, 55(7):1651–1656, 2010.
- [34] N. Michael, J. Fink, S. Loizou, and V. Kumar. Architecture, abstractions, and algorithms for controlling large teams of robots: Experimental testbed and results. *Robotics Research*, pages 409–419, 2011.
- [35] L. Moreau. Stability of multiagent systems with time-dependent communication links. *Automatic Control, IEEE Transactions on*, 50(2):169–182, 2005.
- [36] M. Nieuwenhuisen, R. Steffens, and S. Behnke. Local multiresolution path planning in soccer games based on projected intentions. In *RoboCup 2011: Robot Soccer World Cup XV*, volume 7416 of *LNCS*, pages 495–506. Springer, 2012.
- [37] S. Nouyan, R. Groß, M. Bonani, F. Mondada, and M. Dorigo. Teamwork in self-organized robot colonies. *Evolutionary Computation, IEEE Transactions on*, 13(4):695–711, 2009.
- [38] R. Olfati-Saber and R.M. Murray. Consensus problems in networks of agents with switching topology and time-delays. *Automatic Control, IEEE Transactions on*, 49(9):1520–1533, 2004.
- [39] L.E. Parker. Alliance: An architecture for fault tolerant multirobot cooperation. *Robotics and Automation, IEEE Transactions on*, 14(2):220–240, 1998.
- [40] L.E. Parker. Path planning and motion coordination in multiple mobile robot teams. In *Encyclopedia of Complexity and Systems*, 2009.
- [41] L.E. Parker and Fang Tang. Building multirobot coalitions through automated task solution synthesis. *Proceedings of the IEEE*, 94(7):1289–1305, July 2006.
- [42] M. Pavone, A. Arsie, E. Frazzoli, and F. Bullo. Equitable partitioning policies for robotic networks. In *Robotics and Automation (ICRA), IEEE International Conference on*, pages 2356–2361, 2009.
- [43] M.L. Puterman. *Markov decision processes: Discrete stochastic dynamic programming*. John Wiley & Sons, Inc., 1994.
- [44] M. Quigley, B. Gerkey, K. Conley, J. Faust, T. Foote, J. Leibs, E. Berger, R. Wheeler, and A. Ng. ROS: An open-source robot operating system. In *ICRA workshop on open source software*, volume 3, 2009.
- [45] J. Rashid, M. Broxvall, and A. Saffioti. A middleware to integrate robots, simple devices and everyday objects into an ambient ecology. *Pervasive and Mobile Computing*, 2012.
- [46] M. Riedmiller, T. Gabel, R. Hafner, and S. Lange. Reinforcement learning for robot soccer. *Autonomous Robots*, 27(1):55–73, 2009.
- [47] E. Şahin. Swarm robotics: From sources of inspiration to domains of application. *Swarm Robotics*, pages 10–20,

2005.

- [48] M. Schwarz, M. Schreiber, S. Schueller, M. Missura, and S. Behnke. Nimbro-OP humanoid teensize open platform. In *Proceedings of 7th Workshop on Humanoid Soccer Robots, IEEE-RAS International Conference on Humanoid Robots*, 2012.
- [49] S.L. Smith and F. Bullo. Monotonic target assignment for robotic networks. *Automatic Control, IEEE Transactions on*, 54(9):2042–2057, 2009.
- [50] P. Stone and M. Veloso. Multiagent systems: A survey from a machine learning perspective. *Autonomous Robots*, 8(3):345–383, 2000.
- [51] R.S. Sutton and A.G. Barto. *Reinforcement learning: An introduction*. Cambridge Univ Press, 1998.
- [52] P. Svestka and M. Overmars. Coordinated path planning for multiple robots. *Robotics, IEEE Transactions on*, 23:125–152, 1998.
- [53] H.G. Tanner, S.G. Loizou, and K.J. Kyriakopoulos. Non-holonomic navigation and control of cooperating mobile manipulators. *Robotics and Automation, IEEE Transactions on*, 19(1):53–64, 2003.
- [54] J. Taylor, J. Shotton, T. Sharp, and A. Fitzgibbon. The vitruvian manifold: Inferring dense correspondences for one-shot human pose estimation. In *Computer Vision and Pattern Recognition (CVPR)*, 2012.
- [55] S. Thrun, W. Burgard, and D. Fox. *Probabilistic robotics*. MIT press Cambridge, MA, 2005.
- [56] P. Tournassoud. A strategy for obstacle avoidance and its application to multi-robot systems. In *Robotics and Automation, IEEE International Conference on (ICRA)*, pages 1224–1229, 1986.
- [57] A.E. Turgut, H. Çelikkanat, F. Gökçe, and E. Şahin. Self-organized flocking in mobile robot swarms. *Swarm Intelligence*, 2(2):97–120, 2008.
- [58] N. Vlassis. A concise introduction to multiagent systems and distributed artificial intelligence. *Synthesis Lectures on Artificial Intelligence and Machine Learning*, 2007.
- [59] M. Wooldridge, N. R. Jennings, and D. Kinny. The Gaia methodology for agent-oriented analysis and design. *Journal of Autonomous Agents and Multi-Agent Systems*, 3(3):285–312, 2000.
- [60] K.M. Wurm, C. Stachniss, and W. Burgard. Coordinated multi-robot exploration using a segmentation of the environment. In *Intelligent Robots and Systems, IEEE/RSJ International Conference on (IROS)*, pages 1160–1165, 2008.
- [61] P.R. Wurman, R. D’Andrea, and M. Mountz. Coordinating hundreds of cooperative, autonomous vehicles in warehouses. *AI Magazine*, 29(2):9–19, 2008.
- [62] F. Xiao and L. Wang. Asynchronous consensus in continuous-time multi-agent systems with switching topology and time-varying delays. *Automatic Control, IEEE Transactions on*, 53(8):1804–1816, 2008.
- [63] M.M. Zavlanos, M.B. Egerstedt, and G.J. Pappas. Graph-theoretic connectivity control of mobile robot networks. *Proceedings of the IEEE*, 99(9):1525–1540, 2011.

Manuskripteingang: 11. Oktober; Akzeptanz: 26. November;
Überarbeitung: 21. Dezember 2012.

Prof. Dr. rer. nat. Sven Behnke ist Leiter der Arbeitsgruppe Autonome Intelligente Systeme am Institut für Informatik der Universität Bonn. Hauptarbeitsgebiete: Kognitive Robotik, Maschinelles Lernen, Computer Vision.

Adresse: Universität Bonn, Institut für Informatik VI, Friedrich-Ebert-Allee 144, D-53113 Bonn, Fax: +49-(0)228-73-4425, E-Mail: behnke@cs.uni-bonn.de