

# Einfaches Steuerungskonzept für mobile Roboter in dynamischen unstrukturierten Umgebungen

Thomas Krause, Peter Protzel

Technische Universität Chemnitz  
Fakultät für Elektrotechnik und Informationstechnik  
Institut für Automatisierung  
thomas.krause@etit.tu-chemnitz.de  
peter.protzel@etit.tu-chemnitz.de

**Zusammenfassung.** Es wurde eine Steuerungsarchitektur für mobile Roboter entwickelt, die den Anforderungen einer teilstrukturierten, weitläufigen Umgebung im Außenbereich gerecht wird. Die Architektur ist hierarchisch vernetzt aufgebaut und in verschiedene parallel arbeitende Prozesse aufgeteilt. Dabei wird auch der notwendige Informationsfluss von den Rohdaten bis zu Informationen aus dem Weltmodell betrachtet. Die Architektur wurde auf zwei verschiedenen Roboterplattformen in unterschiedlichen Umgebungen getestet.

## 1 Einleitung

Eine leistungsfähige Steuerungsarchitektur für mobile Roboter ist die Grundlage für ein komplexes Verhalten und eine geschickte Interaktion mit der Umgebung. Eine geeignete Architektur kann helfen, selbst bei komplexeren Aufgaben und Strukturen den Überblick zu behalten und erleichtert die Implementation neuer Elemente. Für den Innenbereich existieren bereits einige Strukturen, die den Bedürfnissen einer vorhersehbaren Umgebungssituation in einer strukturierten Umgebung gerecht werden [3]. Der Einsatzbereich in dynamischen und teilstrukturierten Außenbereichen stellt neue Anforderungen an die Leistungsfähigkeit solcher Strukturen. In diesem Beitrag wird eine diesen Bedingungen angepasste leistungsfähige Steuerungsarchitektur vorgestellt. Dabei werden Testergebnisse auf zwei verschiedenen Plattformen und in unterschiedlichen Umgebungen präsentiert.

## 2 Definition der Einsatzumgebung

Der Roboter soll in einem relativ großen Gelände außerhalb von Gebäuden navigieren. Das Gelände kann als teilstrukturiert betrachtet werden. Dort können geometrisch einfache Objekte, wie Häuser, Container oder Straßen, aber auch komplexe Objekte, wie Autos, Bäume, Sträucher oder Parkbänke vorhanden sein. Neben diesen stationären Objekten können aber auch dynamische Objekte, wie Menschen, Tiere oder fahrende Autos auftreten. Es ist derzeit praktisch

unmöglich alle Fälle, die in so einer Umgebung auftreten können, mit Sensoren zu erfassen oder gar vorher einzuplanen. Hinzu kommt dass ein Roboter auf den unterschiedlichen Untergründen (Sand, Kies, Asphalt, Wiese...) sehr unterschiedlich fährt, was die Erfassung der Bewegungen anhand von Odometrie stark beeinflusst. Daraus ergeben sich einige Anforderungen an die Leistungsfähigkeit der Steuerungsarchitektur.

1. Der Roboter muss schnell auf plötzliche Ereignisse, wie vor dem Roboter auftauchende Objekte (Hindernisse) reagieren können.
2. Der Roboter muss einfache Bewegungen auf unterer Ebene ausführen können, ohne dass eine übergeordnete Instanz, ständig regeln muss.
3. Es müssen dennoch komplexe Bewegungsformen ausführbar sein.
4. Trotz der Komplexität darf die Übersicht nicht verloren gehen.

Es gibt verschiedene Ansätze für die Steuerung von Roboterverhalten. Das Spektrum reicht dabei von Zustandsgraphen über reaktives Verhalten bis zu planenden Instanzen anhand von komplexen Weltmodellen. Diese Ansätze haben ihre spezifischen Vor- und Nachteile und sind in den entsprechenden Situationen sehr effektiv. So kann man mit dem Zustandsgraphen viele Pläne aufbauen und komplexe Verhalten kreieren. Voraussetzung dabei ist jedoch, dass man alle möglichen auftretenden Fälle im Zustandsgraphen abdeckt. Bei größeren Strukturen wird der Zustandsgraph sehr schnell unübersichtlich. Das reaktive Verhalten reagiert schnell und zuverlässig auf plötzliche Ereignisse. Komplexere Verhalten sind damit jedoch schwer umzusetzen. Die Planung anhand von komplexen Weltmodellen ermöglicht sehr komplexes und vorrausschauendes Verhalten. Jedoch ist dieses Verfahren oft sehr langsam und kann auf plötzlich eintretende Ereignisse meist nicht schnell genug reagieren [4].

### 3 Die Steuerungsarchitektur

Angelehnt an die Steuerungsarchitektur eines Reinigungsroboters aus [1], in der eine hierarchische Struktur für eine Reinigungsmaschine für Explorationsaufgaben beschrieben wird, wurde versucht, die Vorteile der verschiedenen Strukturen in einer kombinierten Architektur miteinander zu vereinen. Abbildung 1 zeigt die hierarchisch vernetzte Struktur der Steuerung.

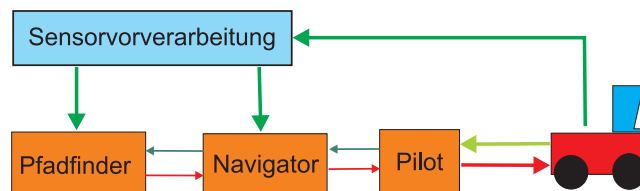


Abb. 1. Struktur der Architektur

Die verschiedenen Anforderungen werden auf verschiedenen Ebenen (Hierarchien) bearbeitet. Dabei nutzt jede Ebene verschiedene Informationen als Eingang. Die Architektur ist von der Aufgabenteilung her mit der einer Besatzung einer Schiffsbrücke vergleichbar. Diese Idee wurde bereits in der Arbeit von [5] aufgegriffen.

### 3.1 Der Pilot

Auf der untersten Ebene arbeitet der Pilot, der als alleiniger Agent Zugriff auf die Antriebe des Roboters hat. Er bekommt als Eingabe vom Roboter die Rohdaten der Sensoren, die Hindernisse direkt erkennen lassen, wie zum Beispiel die Entfernungssensoren. Aus Modulen, die eine Sensordatenvorverarbeitung vornehmen und dem Weltmodell können ebenfalls Informationen zu Hindernissen als Eingabe genutzt werden, insofern sie vorliegen. Der Pilot prüft anhand dieser Informationen und seinem aktuellen Bewegungsbefehl, ob sich ein Hindernis in Fahrtrichtung befindet oder nicht. Je nachdem ob und wie weit ein Hindernis entfernt ist, führt er die gewünschte Bewegung aus oder stoppt den Roboter. Vor dem Hindernis verringert er automatisch die Geschwindigkeit so weit, dass er den Roboter noch vor dem Hindernis zum Stehen bringen kann. Er ist damit rein reaktiv aufgebaut und kann so schnell auf unvorhergesehene Situationen reagieren. Damit ist der Roboter in der Lage gezielte Bewegungen relativ zum Koordinatensystem des Roboters auszuführen und in einfachen Ansätzen auf seine Umgebung zu reagieren.

### 3.2 Der Navigator

Der Navigator in der zweiten Ebene berechnet aus globalen Koordinaten die Bewegungsbefehle für den Piloten, die ausgeführt werden müssen, um den vorgegebenen Zielpunkt zu erreichen. Zusätzlich beinhaltet er eine Regelung, die den Roboter auf einer geraden Linie zum Zielpunkt steuert. Dazu wird bei Empfang eines neuen Fahrbefehls in globalen Koordinaten eine Linie zum Zielpunkt berechnet. Während der Roboter zum Ziel fährt, wird der Abstand zur virtuellen Linie zyklisch berechnet. Resultierend aus dieser Abweichung wird zu dem Richtungsvektor der virtuellen Linie ein Korrekturvektor addiert. Daraus ergibt sich ein neuer Richtungsvektor, der Richtung Linie und Zielpunkt zeigt und den Roboter so wieder auf seinen Pfad bringt. Abbildung 2 verdeutlicht den Ablauf.

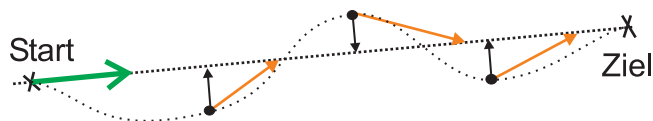


Abb. 2. Bewegung des Roboters auf einer Linie zum Ziel

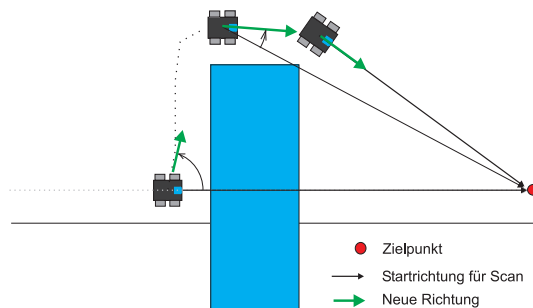
Als Eingang wird nur die globale Position genutzt. Die Regelung, auf einer Linie entlang zu fahren, bringt zwei wesentliche Vorteile:

1. Der Roboter bewegt sich auf einer geraden Linie kontrolliert zum Zielpunkt.
2. Wird die Positionsbestimmung korrigiert, etwa weil neue charakteristische Punkte erkannt wurden, korrigiert sich auch die Bewegung des Roboters automatisch, da sofort Korrekturvektoren auf der Basis der neuen Positionsinformationen berechnet werden.

Damit ist der Roboter in der Lage, im globalen System Bewegungen gerichtet auszuführen und Zielpunkte auf einer Geraden anzufahren. Dabei reagiert er durch den Piloten auf Hindernisse, indem er vor ihnen abbremst und anhält. Erhält der Navigator einen neuen Fahrbefehl, verwirft er den alten und berechnet die virtuelle Linie erneut.

### 3.3 Der Pfadfinder

Der Pfadfinder realisiert das erste komplexere Verhalten des Roboters. Wenn vom Piloten über den Navigator eine Meldung zu einem Hindernis gemacht wird, berechnet er einen Ausweichkurs um das Hindernis, ohne komplexe Informationen seiner Umgebung zu nutzen. Auch er bekommt als Befehl lediglich einen Fahrbefehl. Daraus berechnet er, wie der Navigator, eine virtuelle Linie zum Zielpunkt und gibt anschließend den Fahrbefehl weiter an den Navigator. Taucht auf dem Weg zum Ziel ein Hindernis im Sensorbereich auf und der Pilot meldet, dass er ein Hindernis in seiner Fahrtrichtung erkannt hat, sucht der Pfadfinder nach einer Lücke, durch die der Roboter passt. Aus den gefundenen Lücken wählt er diejenige mit der geringsten Abweichung vom eigentlichen Kurs und fährt dann anderthalb Roboterlängen (Dieser Wert kann auch variieren) in diese Richtung. Danach wiederholt er den Vorgang so lange bis die neue Richtung nicht mehr von der Richtung zum Hindernis abweicht.



**Abb. 3.** kognitive Hindernisumfahrung

Dann gibt er den neuen Fahrbefehl bis zum Zielpunkt. Für den Navigator sind das stets neue Fahrbefehle.

Damit ist der Roboter zusätzlich in der Lage Hindernisse ohne Kenntnis einer globalen Karte zu umfahren und dennoch sein Ziel zu erreichen. In den

darüberliegenden Schichten, können nun weitere Agenten installiert werden, die höhere Aufgaben, wie Pfadplanung oder Exploration realisieren.

## 4 Sensorinformationen

Jede der genannten Schichten benötigt unterschiedliche Sensorinformationen. Für den Piloten müssen die Rohdaten schnell zur Verfügung stehen, während der Navigator nur die Position im globalen System benötigt. Der Pfadfinder nutzt bereits vorverarbeitete Sensordaten und nutzt nur noch die Ergebnisse. Abbildung 4 zeigt wie vernetzt die einzelnen Informationen untereinander und zu den Agenten sind.

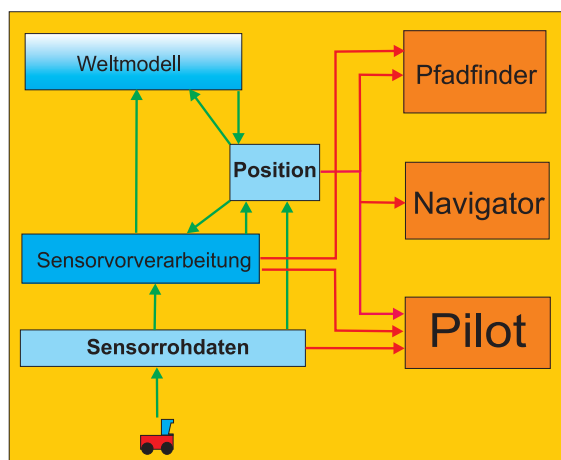


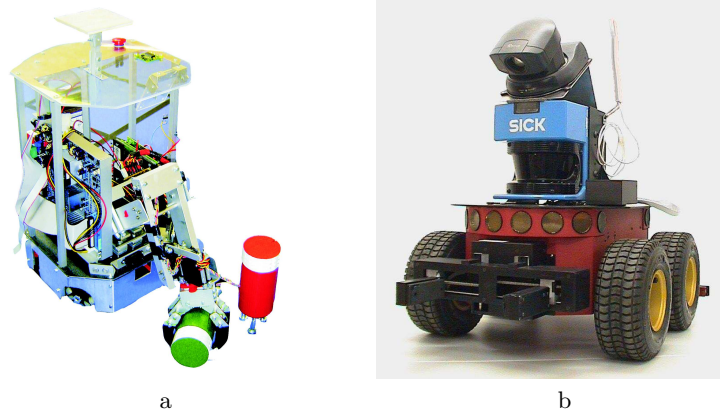
Abb. 4. Informationsfluss in der Steuerungsstruktur

Dieser Informationsfluss ist durchaus noch ausbaufähig. Wichtig bei der Umsetzung der einzelnen Agenten ist die sichere Nutzung der Informationen. Jeder Agent muss für auf fehlende Informationen robust reagieren. So muss der Pilot zum Beispiel anhalten, wenn er keine Sensorinformationen hat. Die Position nimmt dabei eine besondere Stellung ein. Alle Agenten benötigen die Position des Roboters im globalen System und fragen diese ab. Die Positionsbestimmung oder Lokalisierung ist ein eigenes Forschungsgebiet und kann hier nicht weiter betrachtet werden. Hier werden verschiedene Algorithmen angewendet, um eine möglichst genaue Position zu bestimmen. Je genauer sich die Position bestimmen lässt, desto genauer und feinfühlicher wird sich der Roboter mit dieser Struktur verhalten.

Die Daten des Weltmodells werden erst in der nächsten Ebene, zum Beispiel zur Pfadplanung genutzt. Da sie aber für die Bestimmung der Position nützlich sind, ist es bereits mit herangezogen worden.

## 5 Evaluierung

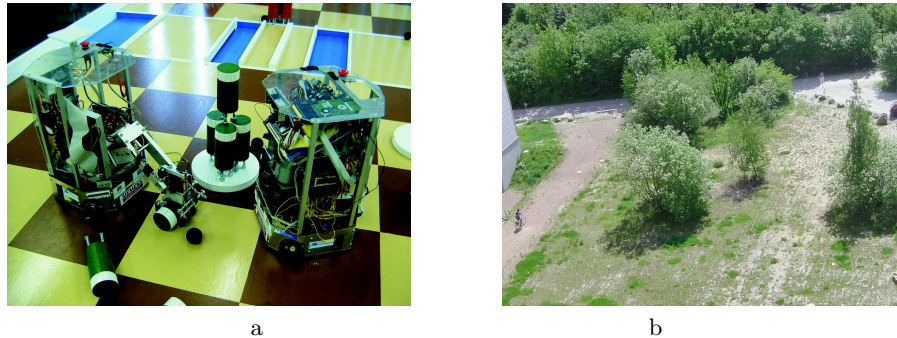
Die vorgestellte Steuerungsarchitektur wurde auf zwei unterschiedlichen Robotersystem in unterschiedlichen Umgebungen getestet. Abbildung 5 zeigt beide Systeme. Der rechte Roboter ist vom Typ Pioneer 2 AT der Firma Active Media [7] (HANS), der im Außengelände operiert und der linke Roboter ist eine Eigenkonstruktion, die ursprünglich für den Roboterwettbewerb Eurobot [6] gebaut wurde und dort unter Wettkampfbedingungen sein Verhalten erfolgreich unter Beweis gestellt hat. Sein Operationsgebiet ist der ebene Innenbereich.



**Abb. 5.** Die Testplattformen: a: VIDAR, Eigenbauplattform mit holonomem Antriebssystem und Greifer; b: HANS, Pioneer 2 AT Plattform mit Sensorerweiterungen.

Abbildung 6 zeigt die beiden Testumgebungen. Vidar fährt auf dem Spielfeld (links), auf dem auch andere Roboter agieren und in dem Objekte (farbige Kegel) zufällig herumliegen und bewegt werden können. Im Außenbereich (rechts) wurde Hans getestet. Im Gelände sind vor allem Menschen als bewegliche Objekte herumgelaufen.

Auf Vidar war als oberste Ebene eine Zustandsmaschine umgesetzt, die verschiedene Aufgaben auf dem Spielfeld erfüllte. Auf Hans war eine Folge von Zielpunkten vorgegeben, die abgefahren werden sollten. Karten und eine komplexere Umweltrepräsentation wurden nicht aufgenommen. Beide Roboter zeigten ein stabiles Verhalten in ihrer Umgebung. Vidar wich stehenden Hindernissen sicher aus und fuhr sicher die von der Zustandsmaschine vorgegebenen Zielpunkte an. Als einziges dynamisches Hindernis fuhr der gegnerische Roboter herum. Diesem wich er immer sicher aus und nahm danach sofort wieder Kurs auf sein eigentliches Ziel. Er zeigte jedoch Schwierigkeiten, wenn der gegnerische Roboter auf ihn zu fuhr und ihn rampte. Da in der Architektur in den unteren Ebenen kein Zurückweichen vorgesehen ist, blieb er nur vor dem Roboter stehen und versuchte zu den Seiten auszuweichen. Wenn der gegnerische Roboter keine Kollisionsvermeidung besaß bzw. den Roboter nicht erkannt hat und weiter fuhr, kam es zur



**Abb. 6.** Die Testgebiete: a: Spielfeld für VIDAR b: Außenbereich für HANS

Kollision. Hier könnte man zusätzlich ein aktives Abstandhalten zum Hindernis einbauen, wodurch der Roboter auch zurückweichen würde. Eine weitere Möglichkeit wäre in einer übergeordneten Instanz ein Verhalten zu implementieren welches den Roboter einen größeren Bogen um bestimmte Hindernisse machen lässt. Hans fuhr seine Zielpunkte im Außengelände sauber ab und wich dabei zwei Felsen und einem Busch problemlos aus und erreichte seine Zielpunkte. Stellte sich ihm ein Mensch plötzlich in den Weg, wich er aus oder hielt an und nahm anschließend wieder Kurs auf sein Ziel. Ein Zielpunkt lag genau innerhalb eines Felsens, was er nicht anfahren konnte. Hier fuhr er bis an den Felsen heran und blieb dort stehen, da er dem Ziel nicht mehr näher kam. Um solche Probleme zu lösen muss eine übergeordnete Planungsinstanz entsprechend eingreifen und neue Zielpunkte vorgeben.

## Literaturverzeichnis

- [1] *Hierarchische Steuerung für einen mobilen Roboter zur autonomen Erkundung seiner Einsatzumgebung*, Thomas Edlinger, 1997, Fortschr.-Ber. VDI Reihe 8 Nr. 638 Düsseldorf, VDI Verlag
- [2] *Pioneer 2 Mobile Robot – Operations Manual*, ActivMedia Robotics, LLC, 2001
- [3] *Intelligentes hierarchisches Regelungskonzept für autonome mobile Robotersysteme*, Jürgen Guldner, 1995, Fortschr.-Ber. VDI Reihe 8 Nr. 509 Düsseldorf, VDI Verlag
- [4] *Behavior-based Robotics*, Ronald C. Arkin, 1998, ISBN 0-262-01165-4, Massachusetts Institute of Technology
- [5] *Konzeption eines Modells zur automatischen Navigation mit einem mobilen Roboter*, Tobias Otto-Adamczak, Diplomarbeit, Chemnitz, 2002
- [6] *Roboterwettbewerb Eurobot*, [www.eurobot.org](http://www.eurobot.org)
- [7] *Activ Media*, [www.activmedia.com](http://www.activmedia.com)