



Hochschule für Angewandte Wissenschaften Hamburg  
*Hamburg University of Applied Sciences*

# Seminarbericht

Carsten Schulz

Kollisionsfreie Routenplanung autonomer  
Fahrzeuge

# Inhaltsverzeichnis

|  |           |
|--|-----------|
| <b>Abbildungsverzeichnis</b>                                       | <b>3</b>  |
| <b>1. Einführung</b>   | <b>4</b>  |
| <b>2. Bestehende Lösungsansätze</b>                                | <b>5</b>  |
| 2.1. Freie Routenplanung . . . . .                                 | 5         |
| 2.1.1. Nicht holonome Fahrzeuge . . . . .                          | 6         |
| 2.1.2. Approximate Cell Decomposition . . . . .                    | 6         |
| 2.1.3. Multiroboter Planungsverfahren . . . . .                    | 7         |
| 2.2. Dezentrale Reservierung von Arbeits-Zeit-Raum . . . . .       | 8         |
| 2.3. Graphenbasierte Routenplanung . . . . .                       | 9         |
| 2.3.1. Friendly Interactive Robot for Service Tasks . . . . .      | 10        |
| 2.3.2. Signaltafelprinzip . . . . .                                | 10        |
| 2.3.3. Fahrerlose Transportsysteme in Containerterminals . . . . . | 11        |
| <b>3. Zusammenfassung</b>  | <b>13</b> |
| <b>4. Simulationsergebnisse und Ausblick</b>                       | <b>15</b> |
| <b>A. Anhang</b>   | <b>16</b> |
| <b>Literaturverzeichnis</b>  | <b>20</b> |

# Abbildungsverzeichnis

|   |    |
|---|----|
| 2.1. Arbeitsraum [Quelle: <a href="#">Latombe (2007)</a> ] . . . . .                    | 5  |
| 2.2. Hindernisraum nicht holonome Fahrzeuge . . . . .                                   | 6  |
| 2.3. Approximate Cell Decomposition [Quelle: <a href="#">Latombe (2007)</a> ] . . . . . | 7  |
| 2.4. Arbeits-Zeit-Raum . . . . .  | 8  |
| A.1. STILL Schubmaststapler FM-X . . . . .  | 16 |
| A.2. SICK 3D-Laserscanner . . . . .   | 16 |
| A.3. Typische Konfliktbereiche . . . . .  | 17 |
| A.4. Multiagent Traffic Management . . . . .  | 17 |
| A.5. FIRST . . . . .  | 17 |
| A.6. Trajektorie Container-Terminal [Quelle: <a href="#">Lehmann (2006)</a> ] . . . . . | 18 |
| A.7. Trajektorie und Blockungsbereiche . . . . .  | 19 |

# 1. Einführung

Unter dem Namen STILL-LASER-NAVIGATOR arbeitet die STILL GmbH in Zusammenarbeit mit Leibniz Universität Hannover, Fachgebiet Echtzeitsysteme, an Systemen zur Automatisierung von Serienflurförderzeugen, die flexibel in unterschiedlichsten Industrieumgebungen eingesetzt werden und dort autonom Transportaufträge bearbeiten können.

Mehrere autonome Fahrzeuge sind dabei in einem gemeinsamen Arbeitsbereich eingesetzt und führen dort Fahraufträge aus. Sie nutzen das gleiche Wegenetz und greifen auf die gleiche Infrastruktur zurück. Dabei kann es zu Konflikten kommen, in dem beispielsweise mehrere Fahrzeuge zeitnah Kreuzungsbereiche zu befahren, Engstellen zu überwinden oder Regale gemeinsam zu bedienen haben. Typische Bereiche sind (vgl. Abbildung [A.3](#)):

- Kreuzungsbereiche
- Fahrbahnverjüngungen
- Einspurige Fahrbahnen
- Sackgassen
- Fahrbahnüberlappungen

Im Rahmen des Masterstudiums an der HAW-Hamburg wird in Zusammenarbeit mit der Still GmbH eine Lösung zur Koordination der autonomen Fahrzeuge erarbeitet.

Dieser Seminarbericht spiegelt einen Teil der Literaturrecherche wieder und soll einen kurzen Einblick in die verschiedene Ansätze der kollisionsfreien Routenplanung autonomer Systeme geben.

## 2. Bestehende Lösungsansätze

In diesem Kapitel werden vorhandene Ansätze und Lösungen zur Routenplanung autonomer Fahrzeuge vorgestellt. Dabei wird zunächst in Abschnitt 2.1 auf allgemeine Ansätze eingegangen, bei denen sich die Fahrzeuge in ihrem Umfeld frei bewegen können und nicht an vorgegebene Fahrspuren gebunden sind. Einen ähnlicher Ansatz wird in Abschnitt 2.2 vorgestellt, der sich mit einer automatisierten Ampelsteuerung beschäftigt.

Im Abschnitt 2.3 werden Systeme vorgestellt, bei denen sich die Fahrzeuge nur auf einer vorgegebener Trajektorie bewegen.

### 2.1. Freie Routenplanung

In diesem Abschnitt wird eine kurze Einführung eines allgemeinen Ansatzes zur Wegfindung autonomer Roboter gegeben.

Bei der freien Routenplanung besitzt der Roboter keine feste Trajektorie, sondern kann sich mittels einer Hinderniskarte frei bewegen. In der Hinderniskarte sind alle dem Roboter bekannten Hindernisse eingetragen. Um die spätere Wegfindung einfacher zu gestalten, werden alle Hindernisse um die Kontur der Roboter erweitert - es entsteht der Arbeitsraum. Abbildung 2.1 zeigt wie das Hindernis B um die Kontur des Roboters A erweitert wird.

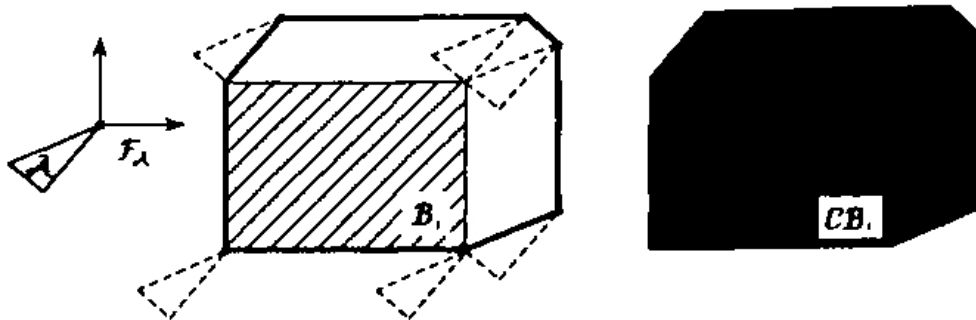


Abbildung 2.1.: Arbeitsraum [Quelle: Latombe (2007)]

Im Arbeitsraum wird der Roboter nun als Punkt betrachtet. Bei diesem Beispiel wird der Einfachheit halber davon ausgegangen, dass sich der Roboter nur in x- und y-Richtung bewegen

kann - Drehungen sind nicht erlaubt. In Abschnitt 2.1.1 wird gezeigt, wie der Arbeitsraum ohne diese Einschränkung erstellt werden kann. Bei den weiteren Betrachtungen wird jedoch von einem punktförmigen Roboter ausgegangen.

In Abschnitt 2.1.2 wird ein Verfahren vorgestellt, mit dem es möglich ist, Wege auf der Hinderniskarte zu finden, indem die Problemstellung auf die Graphentheorie übertragen wird. Der letzte Abschnitt 2.1.3 zeigt Ansätze zur Routenplanung von mehreren Robotern.

### 2.1.1. Nicht holonome Fahrzeuge

Im der Einleitung zu diesem Kapitel wurde davon ausgegangen, dass sich ein Fahrzeug nur in x- bzw y-Richtung bewegen darf und Drehungen nicht erlaubt sind. Einige Fahrzeuge unterliegen zusätzlichen Einschränkungen. So können sich Autos normalerweise nicht seitwärts bewegen. Die Bewegungsfreiheit ist also durch den Lenkwinkel beschränkt. Autos werden deshalb als nicht holonomen Fahrzeugen bezeichnet. Das hat zur Folge, dass sich die Kontur des Fahrzeugs im Hindernisraum in Abhängigkeit des Lenkwinkels ändert. Abbildung 2.2 zeigt die Entstehung des Hindernisraums, der nun wegen der Berücksichtigung des Lenkwinkels um eine Dimension erweitert werden musste.

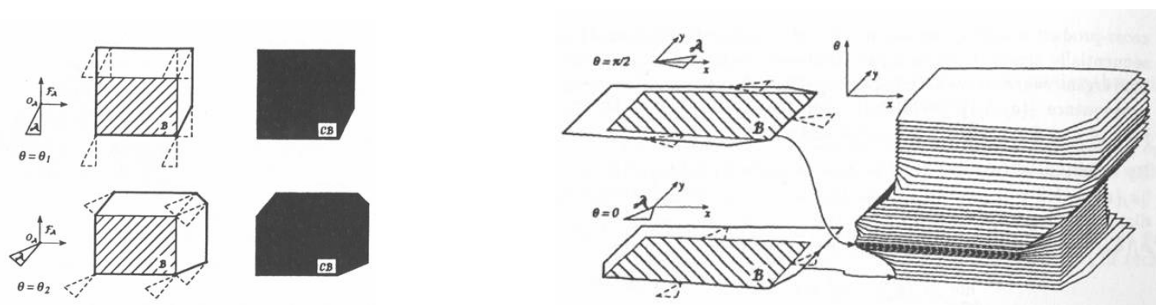


Abbildung 2.2.: Hindernisraum nicht holonome Fahrzeuge

### 2.1.2. Approximate Cell Decomposition

Die Approximate Cell Decomposition ist ein Verfahren, mit dem es möglich ist einen Weg in einem Hindernisraum zu bestimmen. Dabei wird aus dem Arbeitsraum mittels der Quad-Tree-Zerlegung ein Verbindungs-Graph erstellt.

Die Quad-Tree-Zerlegung unterteilt den Raum dabei in Rechtecke. Befinden sich innerhalb eines Rechtecks ein Hindernis, so wird diese Rechteck in vier Rechtecke unterteilt. Dies geschieht so lange, bis sich entweder kein Hindernis mehr innerhalb eines Quadrats befindet oder eine minimale Größe unterschritten wurde.

Abbildung 2.3 zeigt die Quad-Tree-Zerlegung. Freie Felder sind dabei weiss, teilweise mit

Hindernissen belegte Felder grau und Felder, die vollständig mit Hindernissen belegt sind, schwarz markiert.

Der Verbindungs-Graph wird dabei aus den freien weissen Feldern erstellt. Der Weg von  $q_{init}$  nach  $q_{goal}$  kann nun mittels einer Graphensuche, z.B. dem A\*-Algorithmus, gefunden werden.

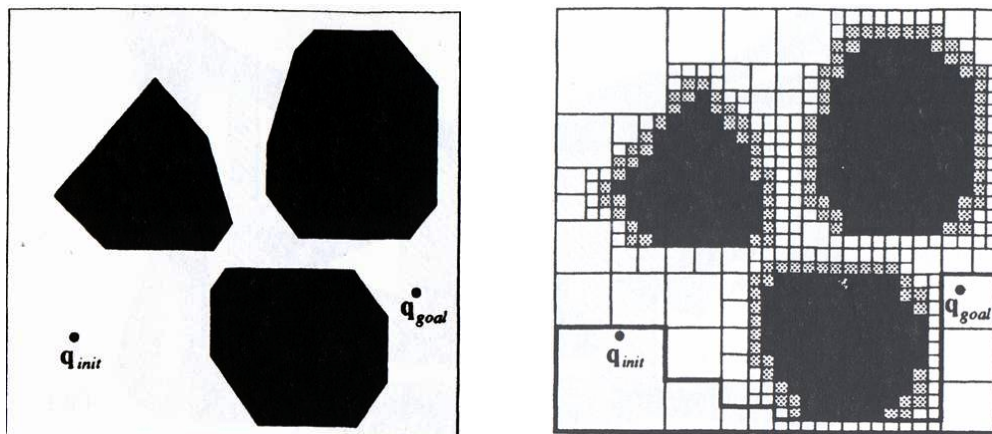


Abbildung 2.3.: Approximate Cell Decomposition [Quelle: Latombe (2007)]

### 2.1.3. Multiroboter Planungsverfahren

In den vorangegangenen Abschnitten wurde eine kurze Einführung in das Themengebiet der freien Wegfindung gegeben. In diesem Abschnitt wird auf einige Herausforderungen der eigentlichen Routenplanung eingegangen.

Der Arbeitsraum muss dafür um eine Dimension erweitert werden, nämlich der Zeit. Daraus entsteht der Arbeits-Zeit-Raum (vgl. Abbildung 2.4). Stationäre Hindernisse behalten ihre Position bei, wobei dynamische Hindernisse ihren Ort in Abhängigkeit von der Zeit verändern können. Bei nicht holonomen Fahrzeugen ist so ein Raum 4-dimensional.

Mit dem Arbeits-Zeit-Raum ist es nun möglich kollisionsfreie Wege für mehrere Fahrzeuge zu planen. Die Planung kann dabei entweder zentral oder dezentral geschehen. Die Zentrale Planung ist dabei weniger komplex als die dezentrale, weil alle Wege zur gleichen Zeit geplant werden können und eine globale Sicht auf alle Route existiert.

Bei der dezentralen Planung werden die Wege der Fahrzeuge nacheinander geplant und anschliessend angepasst. Die Anpassung geschieht unter anderem mittel Velocity-Tuning und Priorisierung gängige Ansätze. Bei Velocity-Tuning werden Kollisionen mit anderen Fahrzeugen durch Veränderungen der Geschwindigkeit verhindert. Die Reihenfolge der Fahrzeuge untereinander kann dabei entweder statisch oder dynamisch festgelegt werden.

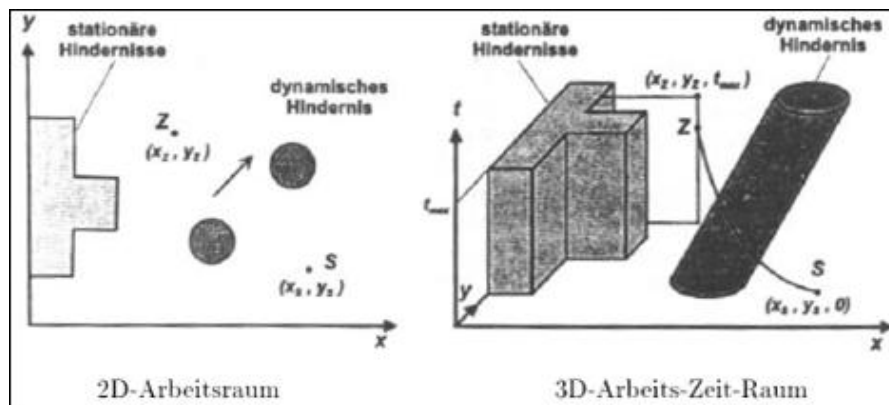


Abbildung 2.4.: Arbeits-Zeit-Raum

Ein ähnlicher Ansatz ist in Abschnitt 2.2 beschrieben, bei dem sich Fahrzeuge Abschnitte im Arbeits-Zeit-Raum reservieren können.

## 2.2. Dezentrale Reservierung von Arbeits-Zeit-Raum

Dressner und Stone (2004) beschreiben ein theoretisches Verfahren zur Verkehrsregelung einer mehrspurigen Kreuzungen für Multiagenten Systeme. Als Basis diente dabei folgende Aussage:

Heutzutage darf ein Auto nur eine Kreuzung überqueren, wenn es das Protokoll erlaubt. Dies geschieht, indem es sich an die roten und grünen Lichter der Ampel hält. In allen anderen Fällen benötigt ein Auto keine zentrale Instanz um Entscheidungen zu fällen.<sup>1</sup>

Die Fahrzeuge müssen sich in der Simulation Zeitschlitze für Überquerung der Kreuzung reservieren. Ausserhalb des Kreuzungsbereichs ist jedoch keine Kontrolle vorgesehen. Der Kreuzungsbereich ist dafür in  $n \times n$  Anforderungsbereiche unterteilt, welche von den Agenten angefordert werden können.  $n$  ist dabei die Granularität des Systems und kann zur Optimierung des Durchsatzes genutzt werden. Pro Anforderungsbereich ist dabei nur ein Fahrzeug zur Zeit erlaubt. Es wird also ein entsprechender Arbeits-Zeit-Raum für den Kreuzungsbereich erstellt.

Eine grundsätzliche Voraussetzung ist, dass Fahrzeuge den Kreuzungsbereich immer sicher verlassen können. Verkehrsstaus hinter der Kreuzung, wie sie in der Realität vorkommen können, werden hier nicht betrachtet. Zudem dürfen die Fahrzeuge die Kreuzung zunächst nur geradeaus überqueren.

<sup>1</sup>nach Dressner und Stone (2004)



Nähert sich ein Fahrzeug der Ampel, meldet es sich dafür bei der Kreuzung an. Dafür übersendet das Fahrzeug mehrere Parameter:

- die Ankunftszeit
- die Geschwindigkeit bei Ankunft
- die Fahrtrichtung bei Ankunft
- die maximale Geschwindigkeit
- die maximale und minimale Beschleunigung
- die Fahrzeuglänge und Fahrzeugbreite

Mittels diesen Parametern simuliert die Kreuzung die Überquerung des Fahrzeugs. Sind alle angeforderten Bereiche zum Zeitpunkt der Durchquerung frei, so erfolgt eine Freigabe für das Fahrzeug. Andernfalls wird die Anforderung abgelehnt. In diesem Fall verzögert der Agent das Fahrzeug und stellt seine Anfrage erneut. Bei einer Freigabe überprüft das Fahrzeug zudem zyklisch, ob es seine Reservierung einhalten kann. Ist es dem Fahrzeug nicht möglich, so storniert es seine Reservierung und stellt eine neue Anfrage.

Die durchgeführten Simulationen haben dabei gezeigt, dass es mit diesem Verfahren möglich ist, einen höheren Verkehrsfluss als bei einer regulären Ampelschaltung zu erreichen. Dies ist darauf zurück zu führen, dass sich mehrere Fahrzeuge gleichzeitig auf unterschiedlichen Fahrspuren und Fahrtrichtungen im Kreuzungsbereich aufhalten können (vgl. Abbildung A.4).

In einer Erweiterung des Verfahrens [Dressner und Stone (2005)] wurden zudem Abbiegevorgänge berücksichtigt und eine verbesserte Anforderungsverarbeitung implementiert. Dabei versucht die Kreuzung, bei einer zunächst abgelehnten Anforderung, eine Lösung durch Anpassung der Fahrzeuggeschwindigkeit zu finden.

## 2.3. Graphenbasierte Routenplanung

Um ein vorhersagbares Verhalten zu garantieren, ist im Gegensatz zur freien Routenplanung die Trajektorie bei graphenbasierter Routenplanung bereits vorgegeben. Ein Verbindungsbaum muss wie bei der Approximate-Cell-Decomposition nicht mehr dynamisch erstellt werden. Analog zu einem Stadtplan oder einer Strassenkarte wird in einer Trajektorie festgelegt wo gültige Wege sind und wie diese Fahrzeugen befahren werden dürfen (siehe Abbildung A.7). Die Umsetzung erfolgt oft mittels gerichteter Graphen. Die Wegfindung erfolgt dabei wie auch bei der Approximate-Cell-Decomposition mittels einer Suche im Graphen.

In den folgenden Unterabschnitten drei verschiedene Lösungsansätze zur kollisionsfreien Routenplanung vorgestellt.

### 2.3.1. Friendly Interactive Robot for Service Tasks

[Causse und Pampagnin \(1995\)](#) beschreiben mit *Friendly Interactive Robot for Service Tasks* (FIRST) ein Transportsystem für schwere Lasten in Krankenhäusern. Die autonomen Fahrzeuge müssen dabei Förderwägen mit Bettwäsche, Müll oder Medikamenten befördern. Mehrere Fahrzeuge teilen sich dabei verschiedene gemeinsamen Bereiche in denen es zu Konflikten kommen kann: Korridore, Fahrstühle Ladestationen, Parkpositionen etc. Die Trajektorie ist bei FIRST als Graph modelliert. Die Kanten des Graphen (Fahrsegmente) sind dabei entweder:

- gerichtet: die Fahrtrichtung ist statisch vorgegeben, oder
- alternierend: die Fahrtrichtung kann sich mit der Zeit ändern

Pro Fahrsegmenten ist nur ein Fahrzeug zur Zeit zugelassen. Die Größe eines Segments entspricht in etwa der Größe eines Fahrzeugs.

Das FIRST-System besteht dabei aus zwei Sub-Systemen: der Bodenstation und den Fahrzeugen. Die Bodenstation ist für die zentrale Verwaltung verantwortlich und kommuniziert per Funk mit den Fahrzeugen. Untereinander können die Fahrzeuge jedoch nicht kommunizieren.

Erhält ein Fahrzeug einen Fahrauftrag von der Basisstation, so führt es ihn selbständig aus. Die Routenberechnung findet dabei dezentral auf dem jeweiligen Fahrzeug, ohne Berücksichtigung der anderen Fahrzeuge, statt. Die benötigten Fahrsegmente muss das Fahrzeug explizit bei der Basisstation reservieren und darf es nur nach erfolgreicher Zuteilung befahren. Erst wenn das Fahrzeug das Segment vollständig verlassen hat, wird es wieder frei gegeben.

Deadlocks werden dabei mittels vordefinierter Regeln vermieden. Fahrsegmente, die sich in kritischen Bereichen befinden, werden markiert und zusammengefasst. Das Befahren eines kritischen Bereichs ist nur erlaubt, wenn alle Segmente des Bereichs frei sind. Die Regeln für die Anforderung und Freigabe der Segmente werden in einer Datenbank hinterlegt.

### 2.3.2. Signaltafelprinzip

[Wang und Premvuti \(1995\)](#) beschreiben ein vollständig dezentrales Verkehrsregelung für mehrere autonome Roboter in einer gemeinsamen Umgebung. Im Gegensatz zu FIRST werden Deadlocks nicht im voraus vermieden, sondern erkannt und behoben.

Vereinfachte gesehen besteht das Streckennetz dabei aus unidirektionalen Fahrsegmenten und Kreuzungen. Pro Segment und auf einer Kreuzung darf sich nur ein Fahrzeug zur Zeit aufhalten. An kritischen Stellen (z.B. Kreuzungen) befinden sich zudem Pufferzonen.

Die Grundidee des Verfahrens besteht darin, dass unter den Fahrzeugen ein Wettbewerb um das exklusive Nutzungsrecht von Fahrsegmenten und Kreuzungen herrscht. Der Wettbewerb und die Zuteilung wird dabei nicht von einer zentralen Instanz gesteuert, sondern geschieht dezentral unter den beteiligten Fahrzeugen.

Zum allgemeinen Informationsaustausch werden Signaltafeln verwendet. Die Signaltafeln werden dabei auf einem zentralen System hinterlegt und enthalten alle relevanten Fahrzeuginformationen. Dazu gehören unter anderem der aktuelle Zustand des Fahrzeugs, die belegten Ressourcen sowie die zukünftig benötigten Ressourcen. Dabei kann jedes Fahrzeug die Signaltafeln der anderen Fahrzeuge abfragen.

Möchte ein Fahrzeug eine Kreuzung  $k$  befahren, werden folgende Aktionen ausgeführt:

1. wenn  $k$  verfügbar ist, bewerbe dich für das exklusive Recht und belege  $k$  bei Erhalt - *Stop*
2. wenn  $k$  blockiert ist, starte die Deadlock-Erkennung  
(Die Deadlockerkennung wird lokal, d.h. nur in der unmittelbaren Umgebung der betroffenen Fahrzeuge, ausgeführt.)
3. wenn kein Deadlock erkannt wurde, bewerbe dich für das exklusive Recht und belege  $k$  bei Erhalt - *Stop*
4. wenn ein Deadlock erkannt: Auflösung mittels Pufferzone - *Stop*

Das Signaltafelprinzip ermöglicht es mit Hilfe einer lokalen Instanz für den Informationsaustausch, eine ansonsten vollständig dezentrale Verkehrsregelung und Konfliktlösung. Für die Abstimmung/Reservierung der Fahrsegmenten wird bei den beteiligten Fahrzeugen das 2-Phase-Commit-Protokoll eingesetzt.

### 2.3.3. Fahrerlose Transportsysteme in Containerterminals

Heutzutage kommen fahrerlose Transportsysteme vermehrt in Containerterminals zum Einsatz. Die Verkehrsregelung ist hier mehr im Bereich der Steuerung als in der Planung anzusiedeln. Es ist zudem kaum möglich sie im Voraus zu planen, da sie in hohem Maße Echtzeitanforderungen unterliegt. Übliche Lösungsansätze basieren auf dem Claiming-Konzept, bei dem die Fahrzeuge einzelne Streckensegmente reservieren. Diese sind dann solange nicht von anderen Fahrzeugen benutzbar, bis sie von dem betreffenden Fahrzeug wieder freigegeben werden. Dabei können zwischen zwei oder mehr Fahrzeugen Deadlocks entstehen, weil sie sich gegenseitig im Fahrkurs blockieren. Es sind also die gleichen Ansätze wie bei FIRST und den Signaltafeln.

Momentan sind in der Praxis meistens die zentrale Steuerungsansätze zu finden. [Lehmann (2006)]. Um die Steuerung zu vereinfachen, kommt in Containerterminals eine stark unidirektionale Trajektorie, die aus mehreren unidirektionalen Kreisen besteht zum Einsatz (siehe Abbildung A.6). Dies ist eine simple Deadlock-Prevention-Strategie, weil ein Zyklus von aufeinander wartenden Fahrzeugen praktisch verhindert wird. Sie funktioniert, weil man den unrealistischen Fall ausschliessen kann, bei dem sich so viele Fahrzeuge auf dem Fahrkurs befinden, dass jedes sofort auf den Vorgänger auffährt.

Bei komplexeren Fahrkursen werden zudem Deadlock-Handling-Strategien eingesetzt. Verfahren zur Deadlock-Erkennung und Lösung kommen dabei weniger zum Einsatz, weil es nur unter großem Aufwand möglich ist, vorhandene Deadlocks wieder aufzulösen, da die Fahrzeuge schon physisch blockiert sind. Zudem ist es wünschenswert unnötige Rangierarbeiten zu unterbinden. Deshalb kommen Algorithmen zur Deadlock-Vermeidung zum Einsatz. Ein bekanntes Beispiel für diese Strategie ist der Banker's Algorithmus. Andere Ansätze verwenden Petri-Netze oder Betriebsmittel-Anforderungsgraphen, um Deadlock vorraus zu sagen und zu vermeiden. Dabei „schaut“ das System eine gewisse Zeit in die Zukunft und prüft auf mögliche Deadlocks. Im Falle eines möglichen Deadlocks wird entweder eine alternative Route berechnet oder das Fahrzeug rechtzeitig angehalten.

Detaillierte Informationen darüber, wie solche Algorithmen wirklich in der Produktion umgesetzt werden, sind nur schwer zu bekommen.

### 3. Zusammenfassung

In dem Kapitel 2 wurden verschiedene Ansätze beschrieben, wie mit dem Problem der kollisionsfreien Routenplanung umgegangen werden kann. Die meisten vorgestellten Ansätze basieren dabei auf Deadlock-Vermeidung, weil bereits entstandene Deadlocks nur unter großem Aufwand wieder aufzulösen, da die Fahrzeuge schon physisch blockiert sind. Meist sind sehr schnell auch andere Fahrzeuge indirekt von dem Deadlock betroffen.

| Kriterien           | Cell Decomposition                               | Arbeits-Zeit-Raum       | FIRST                     | Signaltafeln              |
|---------------------|--|-------------------------|---------------------------|---------------------------|
| <b>Koordination</b> | zentral / dezentral                              | zentral                 | zentral                   | dezentral                 |
| <b>Deadlock</b>     | Avoidance  | Avoidance               | Avoidance                 | Detection                 |
| <b>Navigation</b>   | Hindernisskarte                                  | Zeit-Raum               | Trajektorie               | Trajektorie               |
| <b>Vorteile</b>     | sehr flexibel                                    | gute Bereichsausnutzung | flexible Regelanpassung   | lokale Deadlockbehandlung |
| <b>Nachteile</b>    | Wege nicht Vorhersagbar<br>komplexe Berechnungen | nur simuliert           | evtl. Regeln sehr komplex | benötigt Pufferbereiche   |

Tabelle 3.1.: Gegenüberstellung der Ansätze

Die freie Routenplanung und die Reservierung von Arbeits-Zeit-Raum vermeiden Deadlocks durch gezieltes vorheriges Planen. Dabei können sich die Fahrzeuge sehr flexibel in ihrem Arbeitsraum bewegen. Bei vorgestellten Verfahren wurden nur vereinfachte Fahrzeugmodelle betrachtet. Die Komplexität bei der Berechnung der Wege und der kollisionsfreien Routenplanung nicht holonomer Fahrzeugen erweist sich dabei als großer Nachteil. Zudem ist die Vorhersagbarkeit der Wege nicht gegeben, was eine industrielle Verwendung erschwert.

In dem Kapitel der graphenbasierten Routenplanung wurde gezeigt, dass sich die Aufgabenstellung der Routenplanung auf bekannte Planungsprobleme aus dem Bereich der Betriebssysteme übertragen lässt: der Zuteilung von begrenzten Betriebsmitteln. Die einzelnen Segmente der Trajektorie repräsentieren dabei die einzelnen Betriebsmittel.

FIRST verwendet für die Vermeidung von Deadlock vordefinierte Regeln. Die Regeln könne dabei für die verschiedenen kritischen Bereiche schnell angepasst und zentral in einer Datenbank hinterlegt werden. Eine solche Herrangehensweise ist jedoch nur bei einer relativ einfachen Trajektorie möglich.

Ein völlig anderen Ansatz zur dezentralen Deadlock-Erkennung und Behebung beschreibt das Signaltafelprinzip. Deadlock werden dabei nicht vermieden, sondern lokal erkannt und behoben. Als nachteilig erweisen sich jedoch die benötigten Pufferzonen, weil diese in der Realität auf Grund baulicher Einschränkungen nicht immer umsetzbar sind.

Im Abschnitt [2.3.3](#) wurde schliesslich ein allgemeiner Einblick fahrerloser Transportsysteme in Containerterminals gegeben. Dort geschieht eine Reduzierung der wahrscheinlichkeit von Deadlocks auf Grund der unidirektionalen kreisförmigen Trajektorie. Zudem gibt parallele alternative Strecken zu einem Ziel. Bei einer komplexeren Trajektorie werden Algorithmen zur Deadlock-Avoidance eingesetzt, die z.B. auf dem Banker-Algorithmus oder auf Petrinetzen basieren.

## 4. Simulationsergebnisse und Ausblick

Im Rahmen des Projekts und Seminars wurden durch Simulationen erste Erfahrungen im Bereich der kollisionsfreien Routenplanung gemacht. Dabei wurde eine von der K.U.Leuven Universität entwickelte Simulationsumgebung für fahrerlose Transportsysteme verwendet [Helleboogh u. a. (2006)] verwendet und um benötigte Bestandteile erweitert.

In der Simulationsumgebung bearbeiten mehrere Fahrzeuge autonom Fahraufträge. Die Trajektorie ist dabei als gerichteter Graph modelliert. Im Gegensatz zu den in Abschnitt 2.3 vorgestellten Verfahren, sind pro Fahrsegment grundsätzlich mehrere Fahrzeuge erlaubt. Nur in kritischen Bereichen (wie z.B. Kreuzungen) bedarf es einer Regelung. Die Bereiche werden in der Trajektorie markiert und können aus mehreren Fahrsegmenten bestehen. Pro kritischen Bereich ist maximal ein Fahrzeug erlaubt. In Abbildung A.7 sind die kritischen Bereiche rot hervorgehoben.

Die Verkehrsregelung geschieht, im Vergleich zu den gängigen fahrerlosen Transportsystemen in Containerterminals, nach einem gelockerten Claiming-Konzept: Nur zum Befahren der kritischen Bereiche benötigen die Fahrzeuge eine explizite Freigabe. Alle anderen Fahrsegmente können ohne Freigabe befahren werden, weil die Wege so geplant sind, dass keine Kollision stattfinden kann. Zu vorausfahrenden Fahrzeugen wird durch intelligente Fahrzeugsensorik automatisch Abstand gehalten.

Auf Grund der möglichen Komplexität der Trajektorie erscheint ein regelbasierter Ansatz zur Deadlockvermeidung als nicht sinnvoll. (Zumindest wenn die Regeln manuell erstellt werden müssen.) Ein Ansatz mittels Deadlock-Erkennung ist auf Grund der dafür benötigten Pufferzonen so nicht umsetzbar.

Der Schwerpunkt wurde deshalb auf die dynamische Deadlock-Avoidance gelegt. Erste Versuche mit einem erweiterten Banker-Algorithmus haben gezeigt, dass es möglich ist, die Freigaben dynamisch zu erteilen. Im Rahmen der Masterarbeit soll auf der Grundlage der Deadlock-Avoidance eine zentralgestützte Verkehrsregelung für autonome Fahrzeuge entstehen.

## A. Anhang



Abbildung A.1.: STILL Schubmaststapler FM-X

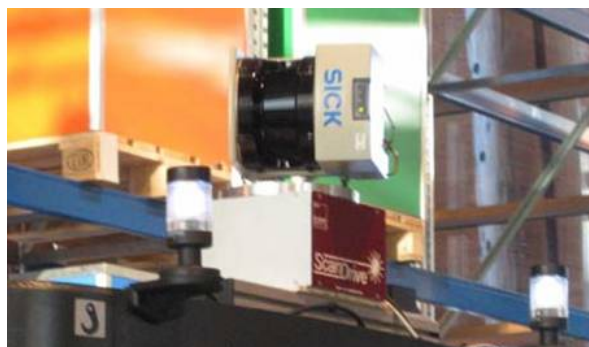


Abbildung A.2.: SICK 3D-Laserscanner



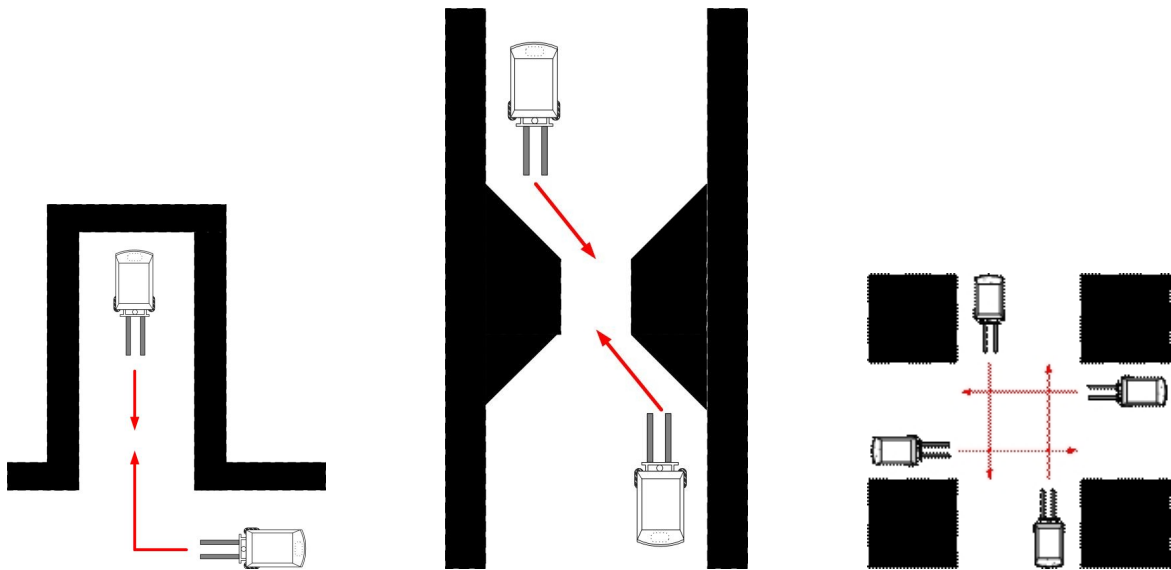


Abbildung A.3.: Typische Konfliktbereiche

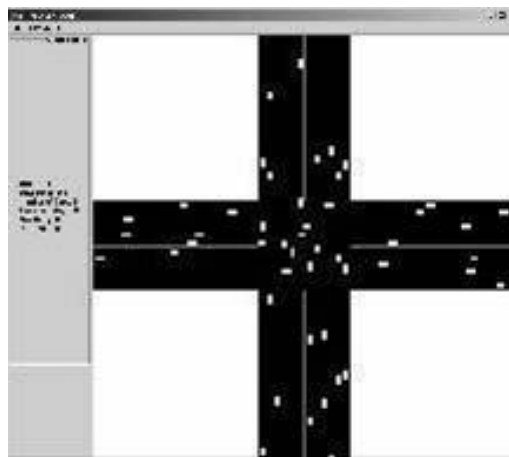


Abbildung A.4.: Multiagent Traffic Management

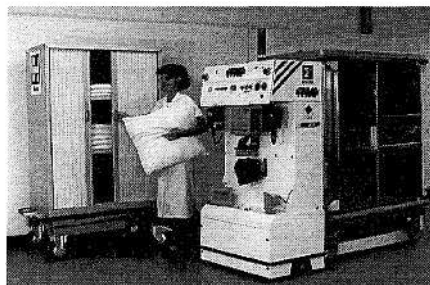


Abbildung A.5.: FIRST

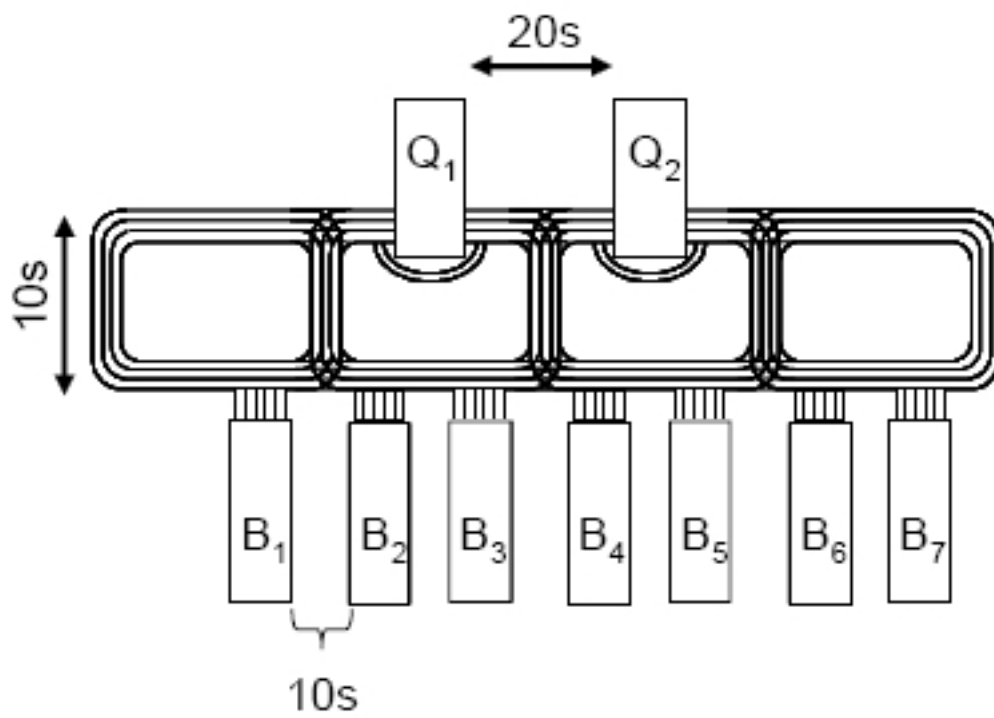


Abbildung A.6.: Trajektorie Container-Terminal [Quelle: [Lehmann \(2006\)](#)]

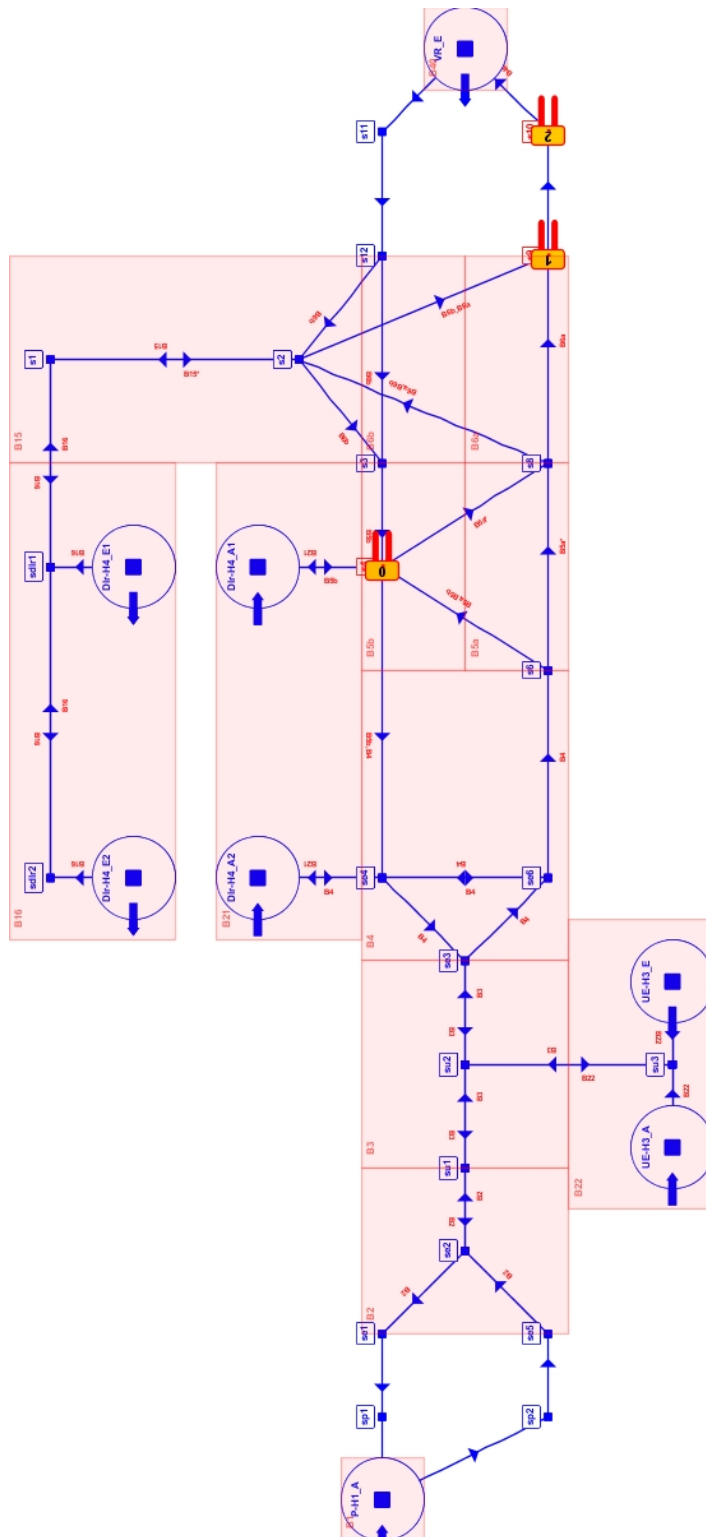


Abbildung A.7.: Trajektorie und Blockungsbereiche

# Literaturverzeichnis

- [Causse und Pampagnin 1995] CAUSSE, O. ; PAMPAGNIN, L. H.: Management of a multi-robot system in a public environment / IEEE. 1995. – Forschungsbericht
- [Dressner und Stone 2004] DRESSNER, K. ; STONE, P.: Multiagent Traffic Management: A Reservation-Based Intersection Control Mechanism / ACM. 2004. – Forschungsbericht
- [Dressner und Stone 2005] DRESSNER, K. ; STONE, P.: Multiagent Traffic Management: An Improved Intersection Control Mechanism / ACM. 2005. – Forschungsbericht
- [Filipp 1996] FILIPP, H. G.: *Kooperative Navigation in einem Multiagentensystem*, Universität Stuttgart, Diplomarbeit, 1996
- [Helleboogh u. a. 2006] HELLEBOOGH, A. ; HOLVOET, T. ; BERBERS, Y.: Testing AGVs in Dynamic Warehouse Environments / ACM. 2006. – Forschungsbericht
- [Latombe 2007] LATOMBE, J.C.: *Robot Motion Planning*. Kluwer Academic Publishers, 2007. – ISBN 0-7923-9129-2
- [Lehmann 2006] LEHMANN, M.: *Einsatzplanung von Fahrerlosen Transportsystemen in Seehafen-Containerterminals*, TU-Berlin, Dissertation, 2006
- [Lüth 1998] LÜTH, T.: *Technische Multi-Agenten-Systeme*. Carl Hanser Verlag, 1998. – ISBN 3-446-19468-1
- [Moorthy und Guan 2007] MOORTHY, K. ; GUAN, W.: *Deadlock Prediction and Avoidance in an AGV System*, Singapore-MIT Alliance, Masterarbeit, 2007
- [Reveliotis 2000] REVELIOTIS, Spyros A.: Conflict Resolution in AGV Systems / School of Industrial and System Engineering. 2000. – Forschungsbericht
- [Tannenbaum 1995a] TANNENBAUM, A. S.: *Moderne Betriebssysteme*. Carl Hanser Verlag, 1995. – ISBN 3-446-18402-3
- [Tannenbaum 1995b] TANNENBAUM, A. S.: *Verteilte Betriebssysteme*. Prentice Hall, 1995. – ISBN 3-930436-23-X
- [Tannenbaum 2007] TANNENBAUM, A. S.: *Distributed Systems*. Pearson, 2007. – ISBN 0-13-239227-5

[Viereck 2001] VIERECK, V.: *Dezentrale kooperative Planung für mobile Robotersysteme*, TU-Berlin, Diplomarbeit, 2001

[Wang und Premvuti 1995] WANG, Jing ; PREMVUTI, Suparek: *Distributed Traffic Regulation and Control for Multiple Autonomous Mobile Robots Operating in Discrete Space* / IEEE. 1995. – Forschungsbericht