

# Bau einer autonomen Räumraupe

Bennett Sattler, Elektro- und Informationstechnik  
Otto-von-Guericke-Universität Magdeburg

**Abstract**— In der heutigen Zeit werden mehr und mehr Arbeitsschritte von selbstständigen Robotern übernommen um die Produktion zu beschleunigen und menschliche Arbeitskräfte einzusparen. Aus menschengesteuerten Maschinen werden so autonome Roboter. Wir haben „die Räumraupe“, das Modell eines autonomen Räumfahrzeuges entwickelt, da diese später in der Praxis in Katastrophenfällen helfen können Straßen für andere Hilfsfahrzeuge wieder befahrbar zu machen. In unserer Arbeit ging es nun darum auf Grundlage der Fuzzy-Logik die „Räumraupe“ die Linie verfolgen zu lassen. Sollten sich nun auf diesem Weg Hindernisse befinden so erkennt der Roboter sie rechtzeitig, nähert sich ihnen langsam und testet dann ob er sie verschieben kann oder nicht. Sollte ersteres der Fall sein dann dreht sich der Roboter um 180° und nutzt die Schiebevorrichtung um das Hindernis von der Fahrbahn herunter zu schieben.

Als Bausatz nutzen wir das „Lego Mindstorms Education Set“ und entwickelten das dazu gehörige Programm in Matlab. Um die Kommunikation zwischen Matlab und dem Lego-NXT Baustein zu gewährleisten nutzten wir ein Matlabpaket der RTWH Aachen.

Als Ergebnis erhielten wir ein funktionstüchtiges Modell und noch einige Anmerkung, auf Basis unseres Entwicklungsprozesses, die erfüllt werden sollten um das Räumfahrzeug in der Praxis nutzen zu können. Zum einen wären es zusätzliche Sensoren um Fußgänger oder andere Dinge zu erkennen die keine Hindernisse sind und zum anderen wäre es eine Satellitensender um zu erkennen welche Fläche bearbeitet werden muss und wo zum Beispiel die Erde hingebacht werden kann.

**Schlagwörter**—Autonomes Fahren, Fuzzy-Logik, Lego Mindstorms, Matlab, Räumfahrzeug

## I. EINLEITUNG

Der Einsatz von Robotern in Arbeitsbereichen die relativ leicht automatisiert werden können ist zum Beispiel in der Automobilproduktion nichts neues mehr. Da es aber noch andere simple Tätigkeiten gibt die durch Roboter ausgeführt werden könnten um den Menschen für wichtigere beziehungsweise kompliziertere Tätigkeiten freizustellen, besteht in dieser Richtung noch Handlungsbedarf.

Als mögliches Beispiel sei das Freiräumen von Straßen aufgeführt. Das ist eine, für den Menschen eintönige aber dennoch wichtige Tätigkeit. Hier liegt einer guter Ansatz für den Roboter vor, da die Entwicklung für autonom fahrende Autos schon fortgeschritten ist, bedarf es hier theoretisch nur ein paar Zusätze um aus einem autonomen Auto ein autonomes Räumfahrzeug zu machen.

Genau dort setzt das Projekt „Räumraupe“ an, welches vom 12.02. bis zum 23.02 im Lego Mindstorms Seminar bearbeitet wurde. Ziel war es solch einen Prototypen modellhaft zu erschaffen und damit ein autonomes Räumfahrzeug herzustellen. Dieses Fahrzeug soll, während es einer Linie folgt (als vereinfachte Straße) selber Hindernisse erkennen und testen ob diese, für das Fahrzeug selber, verschiebbar sind oder umfahren werden müssen. Als Hindernisse dienten leere Pappschachteln und beschwerte Dosen, die zum Beispiel als Geröll oder Ähnliches gesehen werden könnten. Der Roboter wird mithilfe des NXT-Education Sets aufgebaut und ausgeführt. Die Programmierung erfolgt über Matlab und von der RTWH-Aachen wurde ein Paket bereitgestellt um die Kommunikation zwischen Matlab und dem NXT-Baustein realisieren zu können. So konnten einzelne Motoren- und Sensoreingänge angesteuert werden.

## II. VORBETRACHTUNGEN

### A. Linienverfolgung

Als Grundlage zur Linienverfolgung diente die Fuzzy-Logik [1] um einen flüssigeren Ablauf zu garantieren und so die Fehlerhaftigkeit zu reduzieren. Der Unterschied zur herkömmlichen Linienverfolgung ist, dass nicht nur zwischen schwarz und weiß unterschieden wird, sondern auch Graustufen hinzugezogen werden (siehe Abbildung 1). Somit kann schneller und feiner auf Abweichungen reagiert werden und daher wurde dieses Verfahren von uns genutzt.

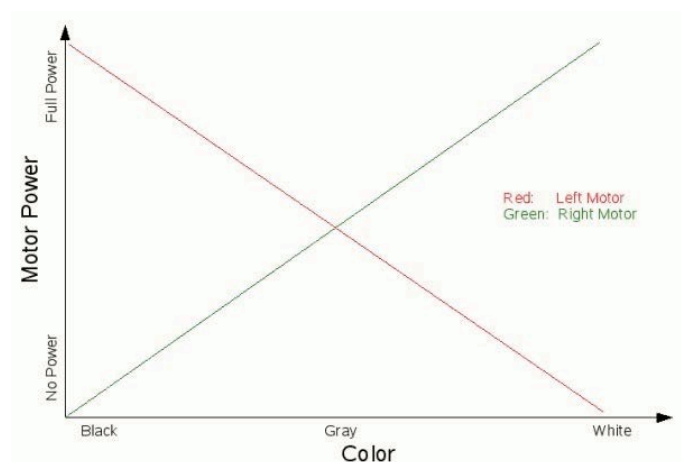


Abbildung 1: Motorenkraft in Abhängigkeit vom Farbwert

Zu beachten ist aber, dass bei mittelmäßigen Lichtverhältnissen in Räumen der Schattenwurf die Lichtsensoren leicht verwirrt und die Linienverfolgung dadurch beeinträchtigt wird.

### B. Grundlegende Programmierung

Zur grundlegenden Programmierung wurde die Befehlsübersicht [2] genutzt. In dieser sind neben der Kommunikation zwischen Matlab und dem NXT-Baustein, auch die grundlegenden Programmabläufe, sowie das Einbinden und Nutzen von Funktionen beschrieben. Was aber während des Erarbeitungsprozesses auffiel, war das der Roboter sich oft in einer Schleife fest fuhr und das Problem nicht gefunden werden konnte, da auch kein Error ausgegeben wurde. Es musste eine Maßnahme zur Ausgabe der aktuell ablaufenden Prozesse gefunden werden.

### III. HAUPTTEIL

Als grundlegendes Konzept lag ein Fahrzeug vor welches eine Differenziallenkung besitzt um ein das Manövrieren leichter und die Wendekreise kleiner zu machen. Des weiteren wurde auf einen kompakten, da robusten Bau geachtet denn ein niedriger Schwerpunkt macht das Verschieben von Gegenständen leichter und verringert die Gefahr dass das Fahrzeug umkippt und beschädigt wird. Dies wird in bei diesem Modell nicht der Fall sein, wird aber in der Praxis von Bedeutung sein, wenn der Roboter über unebenes Gelände fahren muss.



Abbildung 2: Seitenansicht der Räumraupe

In Abbildung 2 ist zu sehen, dass alles sehr nah am Untergrund gehalten ist, um den oben genannten Problemen aus dem Weg zu gehen. Nach dem dies festgelegt war folgte als erste Aufgabe, das Fahrzeug die Linie verfolgen zu lassen, da die restlichen Problemstellungen darauf aufbauten.

Wie in Punkt II.A bereits beschrieben programmierten wir nach der Fuzzy-Logik ("verschwommene Logik"). Aber um dem Schattenwurf und den damit einhergehenden Beeinträchtigungen aus dem Weg zu gehen ließen wir den Lichtsensor (in diesem Fall ein "umfunktionierter" Farbsensor) selber ein blaues Licht ausstrahlen und konnten so "schattenfrei" die Farbwerte messen. Um den einzelnen Farbwerten Motorkräfte zuzuordnen, nahmen wir zwei Testwerte (die schwarze Linie und der weiße Untergrund) und ordneten diese jeweils der Motorkraft 0 dem einen und der Motorkraft 54 dem anderen Motor zu. (siehe Anhang 1)

Weiterhin sollten sich die Motorkräfte gegensätzlich und gleichmäßig verändern. Das hat zur Folge, dass bei einem sehr

hellen beziehungsweise sehr dunklen Untergrund stärker gelenkt wird als bei einem Mittelwert. Der Roboter versucht nun auf der Kante zwischen weiß und schwarz zu fahren. Gegenüber der herkömmlichen Variante wird die Unterlage angestrahlt und verhindert so den größten Teil von Fehlmessung und sollte daher auch später in die Praxis übernommen werden.

Danach ging es darum den Roboter Hindernisse erkennen zu lassen und später diese auf Verschiebbarkeit zu testen. Dafür sind folgende NXT-Sensoren wichtig:

- zwei Ultraschallsensoren
- ein Farbsensor (zur Linienverfolgung)
- ein zusätzlicher Motor
- ein Tastsensor



Abbildung 3: Draufsicht der Räumraupe

Die Ultraschallsensoren befinden sich jeweils unter dem Tastsensor (Abbildung 3 rechte Seite) und einmal seitlich zur Fahrtrichtung (Abbildung 3 untere Mitte). Erstere dient dazu um zu erkennen ob sich ein Hindernis auf der Fahrbahn befindet und letzterer dient wird benötigt um zu erkennen wann der Roboter das Hindernis mit einer Rechtskurve umfahren kann oder ob es sich immer noch auf der Linie befindet.

Der Farbsensor dient wie schon beschrieben zur Verfolgung der Linie und befindet sich daher vorne in der Mitte des Fahrzeugs (Abbildung 3 zwischen dem Tastsensor und dem NXT-Baustein).

Für die Schiebevorrichtung wird noch ein zusätzlicher Motor benötigt um das Objekt nicht nur von dem Roboter vor sich her schieben zu lassen, sondern auch aus dem Weg, das heißt von der Linie wegzuschieben. (siehe Anhang 2)

Das wird dadurch erreicht, dass ab einem bestimmten Punkt im Prozess der Motor aktiviert wird und dadurch die Schiebevorrichtung etwas anstellt, d.h. nicht mehr senkrecht zur Fahrtrichtung, sondern in einem steileren Winkel, sodass das Objekt von der Linie gedrängt wird.

Als letztes wird noch ein Tastsensor benötigt um zu testen ob das Objekt verschiebbar ist oder nicht (darauf wird später genauer eingegangen). Dieser befindet sich vorne über dem Ultraschallsensor (Abbildung 3 rechte Seite). Die Vorrichtung wurde so gebaut das er sich schwerer auslösen lässt, dies

wurde erreicht indem wir zwischen der Kontaktplatte (die mit dem Hindernis in Berührung kommt) und dem Tastsensor ein Bauteil aus Gummi einbauten, welches zusätzliche Kraft erfordert um eingedrückt zu werden und letztendlich diese Kraft weiter auf den Tastsensor zu übertragen und diesen damit auszulösen. Der Aufbau wird in der Abbildung 5 noch etwas genauer dargestellt.

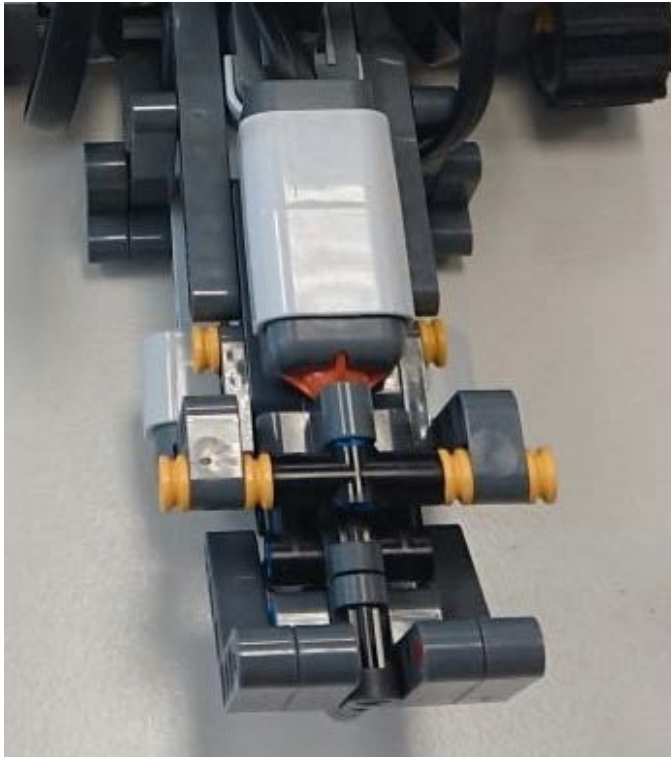


Abbildung 5: Tastsensor

Nachdem der Bau des Fahrzeugs im groben abgeschlossen war ging es an die Programmierung, da es unpassend wäre den ganzen Quellcode zu erklären, wird nur der Inhalt des Quellcodes anhand eines normalen Testablaufes des fertigen Prototyps der "Räumraupe" erklärt.

Die Räumraupe fährt auf das Hindernis hinzu und sobald der Ultraschallsensor eine Wand erkennt hält diese an und dreht sich um ca. 45° nach links und rechts und misst die Entfernung in kleinen Abschnitte. An dem Punkt an dem die niedrigsten Entfernung gemessen wurde bleibt der Roboter stehen, da er nun senkrecht im Bezug zum Hindernis steht (die Strecke ist im geometrischen Sinn das Lot). Daraufhin fährt der Roboter langsam geradeaus mit gleichbleibender Geschwindigkeit. Sollte innerhalb eines kurzen Zeitfensters nicht der Tastsensor ausgelöst werden (das Zeitfenster reicht aus das Hindernis zu erreichen und kurz dagegen zu fahren), fährt die Räumraupe zurück und dreht sich um 180° um den Schieber nutzen zu können. Nun fährt der Roboter geradeaus (unabhängig von der Unterlage) und verdreht dabei langsam den Schieber um das Hindernis aus dem Weg zu schieben. Nach einer bestimmten Zeit fährt er wieder ein Stück zurück bis die Linie gefunden wurde. Nun dreht sich die Räumraupe wieder um 180° zurück in ihre Ausgangsposition und verfolgt die Linie.

Ist das Objekt nicht verschiebbar so umfährt der Roboter das Hindernis in dem er um 90° (nach links) wendet und solange in diese Richtung fährt bis der Ultraschallsensor einen viel größeren Entfernungswert ausgibt. Dann dreht sich der Roboter um 90° in die andere Richtung (rechts) und dies wird solange wiederholt (immer nach rechts) bis der Farbsensor die Linie wieder findet er sich nun dann wieder um 90° dreht und der Linie weiter folgen kann.

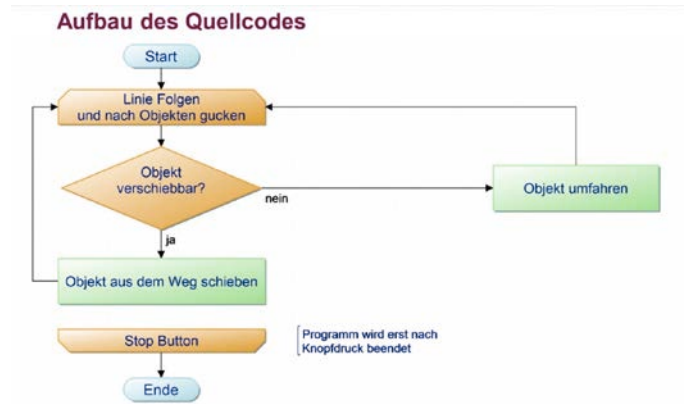


Abbildung 6: Aufbau des Quellcodes

#### IV. ERGEBNISDISKUSSION

Herausgekommen ist ein autonomer Roboter der selbstständig seinen Weg findet und den vorgegebenen Weg (soweit ihm möglich) von Hindernissen befreit. (Beispiel: siehe Abbildung 7)

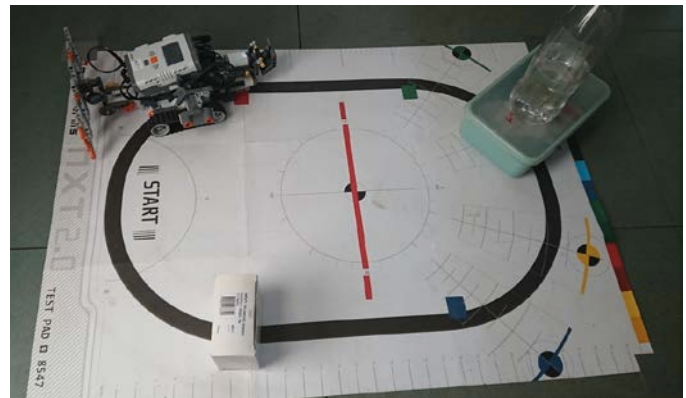


Abbildung 7: Vorführungsaufbau

Als größten Probleme stellten sich 3 Sachverhalte dar.

1. Ungenauigkeiten der Motoren und Sensoren
2. Unklarheiten des aktuellen Ablaufs
3. Umfahren der Hindernisse

Zu 1.: Die Sensoren und Motoren arbeiteten meist nicht so genau wie gewünscht von daher musste, zum Beispiel bei einer Wende des Fahrzeugs die Anzahl der Umdrehungen für jeden Fall spezifisch durch Testen herausgefunden werden und konnte nicht einfach aus der Theorie (sowie es hätte eigentlich möglich sein sollen) umgesetzt werden. Selbiges galt auch für das Messen von Farbwerten oder der Entfernung.

Zu 2.: Anfangs kam es regelmäßig dazu dass der Roboter sich in bestimmten Schleifen "festfuhr" und so Prozesse immer wieder ausgeführt wurden (zum Beispiel: „suche Wand“). Das dies keinen Error in Matlab ergab war es schwer in kurzer Zeit fehler zu finden und zu beheben. Deshalb bauten wir in der GUI („Graphical User Interface“) ein Textfeld ein welche die aktuell laufende Prozesse ausgab, das heißt dass die einzelnen Schleifen beziehungsweise Prozesse benannt werden mussten.

Zu 3.: Durch die beiden vorhergehenden Probleme erwieß sich diese Aufgabe als äußerst schwierig. Denn neben der logischen Abfolge musste noch der Wenderadius beachtet werden und nur durch häufiges Testen konnte schließlich das Wenden um genau 90° absolviert werden.

### V. ZUSAMMENFASSUNG UND FAZIT

Mit dem Verwenden der Fuzzy-Logik als Grundlage zur Linienverfolgung entstand ein autonomer Roboter der anhand eines bestimmten Ablaufes von Fragen und Aufgaben (siehe Abbildung 6) einen gekennzeichneten Weg frei von Hindernissen räumt. Sollte ihm dies nicht möglich sein findet er selbstständig einen Weg und setzt danach seinen alten Weg auf dem gekennzeichneten Abschnitt fort.

Dieser kann als grundlegender Prototyp für autonome Straßenreinigungsfahrzeuge gelten, insofern im Alltag zusätzliche Probleme wie Verkehr oder Fußgänger beachtet werden. Ein wichtiger Faktor ist unserer Meinung nach noch dass außerdem in der Praxis nicht einfach das Hindernis von der Straße auf den Fußgängerweg oder die andere Fahrbahn geschoben werden. So wäre es eine Möglichkeit das Fahrverhalten zu verbessern indem man mithilfe der Satellitentechnik den Verkehr mit einbezieht, mehrere Sensoren anbringt um Fußgänger oder Tiere zu erkennen.

Für die Entscheidungsfindung sollte außerdem noch ein Weg gefunden werden um klar zwischen einem Störfaktor (Äste oder Geröll) und einem normalem Hindernis (Auto oder Motorrad). Des weiteren wäre es nötig in Städten den gefundenen „Dreck“ von der Räumraupe tatsächlich entfernen zu lassen und nicht nur an den Rand zu schieben. Daher wäre ein besserer Einsatz ein Bereich in dem lediglich das bereinigen der Straße gilt und keine Verkehr im Weg steht (zum Beispiel nach einem Erdbeben auf einer Landstraße).

### ANHANG

Aus formatierungstechnischen Gründen sind hier noch zwei Abbildungen angefügt:

	A	B	C
1	Lichtwert	Speed-A	Speed
2			
3	27	0	54
4	36	2	52
5	45	4	50
6	54	6	48
7	63	8	46
8	72	10	44
9	81	12	42
10	90	14	40
11	99	16	38
12	108	18	36
13	117	20	34
14	126	22	32
15	135	24	30
16	144	26	28
17	153	28	26
18	162	30	24
19	171	32	22
20	180	34	20
21	189	36	18
22	198	38	16
23	207	40	14
24	216	42	12
25	225	44	10
26	234	46	8
27	243	48	6
28	252	50	4
29	261	52	2
30	270	54	0

Anhang 1: Lichtwerte und dazugehörige Motorenkraft



Anhang 2: Schiebevorrichtung

### LITERATURVERZEICHNIS

- Stephan Bracher, "NXT Linienfolger programmiert in Fuzzylogik," <http://www.stefans-robots.net/de/nxt-linienfolger-in-fuzzy-logik.php> (Stand: 22.03.18)
- FEIT OVGU (2018, Februar) [Online]. Unterlagen zum Lego Mindstorms Praktikum in der FEIT