

Autonom fahrender Aufklärungsroboter

Gia Bao Truong, Elektromobilität
Otto-von-Guericke-Universität Magdeburg

Abstract— Roboter werden eingesetzt, um Menschen in allen Bereichen des Lebens zu helfen und Probleme zu lösen. In Situationen in denen das Leben eines Menschen gefährdet werden könnte, soll ein Roboter diese Aktion übernehmen, um das Risiko zu verringern. Im Rahmen des Projektseminars 2021 haben wir uns zur Aufgabe gesetzt einen Aufklärungsroboter aus Lego zu entwickeln, der in realen Szenarien eingesetzt werden kann um eine gefährliche Umgebung abzusuchen. In dem folgendem Paper geht es um die schrittweise Entwicklung eines Roboters, der eingesetzt wird um autonom einen Parkour zu bewältigen und währenddessen Informationen über diesen zu sammeln, welche eine Bergungsmission in dem zuvor unerforschten Gebiet zu ermöglichen. Der Einsatz des Roboters kann in einer realen Situation die Gefährdung von menschlichem Leben vorbeugen.

I. EINLEITUNG

Technologie wird in einem rasantem Tempo weiterentwickelt und soll das menschliche Leben immer mehr vereinfachen. In allen Bereichen des Alltags findet man Maschinen, die Arbeit übernehmen können, ob es die Kaffeemaschine ist, die auf Knopfdruck einen heißen Kaffee in die Tasse füllt, oder Drucker, die innerhalb von Sekunden ganze Seiten mit Text füllen können. Autonome Systeme finden heutzutage immer mehr Gebrauch in Fahrzeugen. Dabei können sie helfen mit Kameras und Sensoren sicherer einzuparken, schalten automatisch die Scheinwerfer oder Scheibenwischer an wenn sie benötigt werden, oder übernehmen sogar komplett das Fahren. Solche Assistenzsysteme haben schon vielen Leuten Zeit und Nerven gespart. Autonome Systeme werden jedoch nicht nur benutzt um den Alltag zu vereinfachen, sondern auch um dem Verlust von menschlichem Leben zu umgehen. Ein Bergungsroboter kann dabei helfen in einem Gefahrengebiet einen besseren Überblick zu schaffen um somit die darauf folgenden Handlungen besser planen zu können. Dadurch verhindert der Roboter, dass sich ein Mensch in Gefahr begeben muss, um dieses Gebiet zu erkunden, wodurch auch zusätzlich Zeit gespart werden, welche bei der Rettung von verletzten Personen wichtig ist.

Das Ziel war einen Roboter zu konstruieren, der selbstständig einen Parkour durchfahren soll. Dabei muss er Hindernissen ausweichen, bestimmte Stellen als Gefahr erkennen und umgehen und bei der Opferstelle ein akustisches Signal abgeben. Währenddessen sollen Informationen gesammelt werden, um im Anschluss eine Karte vom Parkour erstellen zu können. Hierbei soll der Roboter aus dem zur Verfügung gestelltem Lego-Mindstorm-Set gebaut und mit dem Programm Matlab angesteuert werden.

II. VORBETRACHTUNGEN

A. Grundtypen von Bergungsrobotern

Man unterscheidet zwischen zwei Typen von Bergungsrobotern. Der erste Typ sind die Aufklärungsroboter. Sie dienen der Aufklärung von gefährlichen Umgebungen mit dem Ziel einen möglichst genauen Überblick über die Umgebung mit ihren Gefahren zu erhalten. Dadurch soll eine darauf folgende Operation in diesem Gebiet effektiver geplant werden können. Der zweite und komplexere Typ sind die Bergungsroboter. Sie sollen zusätzlich zum ersten Typen auch verletzte Personen aus diesem Gebiet transportieren können. Aufgrund der begrenzten Zeit und unserem geringem Vorwissen haben wir uns für den ersten Typen entschieden.

B. Umsetzung

Unser Roboter soll autonom durch einen aufgebauten Parkour bewegen. Dabei soll er sich immer an der Wand rechts von ihm entlangfahren bis er an das Ende des Parkours gelangt. Es wurde sich bei dem Roboter für ein Kettenfahrzeug mit zwei Ketten, die jeweils von einem Motor angetrieben werden entschieden, weil somit Drehungen leichter zu realisieren sind. Über einen Ultraschallsensor, der den Abstand misst, soll der Roboter die Information, darüber wie weit der Roboter von der Wand rechts von ihm entfernt ist, bekommen und sich mit dem Ziel den Abstand möglichst konstant zu halten entsprechen bewegen. Hierbei müssen wir vor dem Start einen Abstand definieren und wenn der gemessene Abstand größer ist als der vordefinierte, dann soll der Roboter sich näher an die Wand bewegen und wenn der gemessene Abstand kleiner ist als der vordefinierte Abstand, dann soll der Roboter sich von der Wand wegbewegen. Der Abstandsausgleich wird durch das Drehen des Roboters, welches durch das Antreiben der Ketten in unterschiedliche Richtungen realisiert wird, ermöglicht. Um zu verhindern, dass sich der Roboter die ganze Zeit nur hin und her dreht gibt es einen Toleranzbereich in dem der Abstand sein darf und in dem der Roboter geradeaus fährt.

C. Fuzzy Logic

In der einfachen Logik wird nur zwischen den zwei Zuständen 0 und 1 beziehungsweise falsch und wahr unterschieden. Für das Bewältigen des Parkours in einer moderaten Zeit reichen zwei Zustände allerdings nicht aus. Entweder könnte der Roboter nur auf einen passenden und unpassenden Abstand, oder auf einen zu kleinen oder zu großen Abstand reagieren. Im ersten Fall fehlt dem Roboter die Information, ob er beim unpassenden Abstand sich der Wand nähern oder sich von der

Wand entfernen soll. Im zweiten Fall würde der Roboter sich nur nach links oder rechts drehen und um den gewählten Abstand schwanken ohne dabei vorwärts zu kommen. Die Fuzzylogik kann zwischen mehr als nur zwei Zuständen unterscheiden und wurde entwickelt um mit Unschärfen zu arbeiten. Übertragen auf unseren Roboter kann der Roboter nun auf mehr als nur zwei Fälle reagieren. Ohne die Fuzzylogik konnte er nur auf einen zu großen, oder einen zu kleinen Abstand reagieren. Mit der Fuzzylogik ist es ihm möglich einen Abstand in >>viel zu groß<<, >>ein wenig zu groß<<, >>passend<<, >>ein wenig zu klein<<, und „zu klein“ zu unterteilen und dem entsprechend einzuschätzen wie er zu reagieren hat. Dies ist wichtig, weil der Roboter nun weiß, ob er seine Fahrt nur leicht korrigieren muss, oder sogar komplett abbiegen muss. Entsprechend zum Abstand kann der Roboter schließlich in beide Richtungen seinen Abstand nur leicht korrigieren, in beide Richtungen abbiegen und bei einem optimalen Abstand geradeaus fahren.

III. REALISIERUNG

A. Aufbau

Der Roboter ist so aufgebaut, dass zwei Lego NXT Motoren direkt nebeneinander befestigt wurden. Die Motoren treiben jeweils eine Kette an und können unabhängig voneinander angesteuert werden. Die Unabhängigkeit beider Ketten wird benötigt um Drehungen im Uhrzeigersinn oder gegen den Uhrzeigersinn zu realisieren. Wenn der Roboter sich im Uhrzeigersinn drehen soll, dann dreht der Motor auf der linken Seite nach vorne und der Motor auf der rechten Seite nach hinten. Bei einer Drehung gegen den Uhrzeigersinn drehen sich die Motoren dementsprechend anders herum. Auf diesen Motoren wird der NXT Stein gesetzt. Der NXT Stein ist die Schnittstelle zwischen dem Roboter und dem Computer. Per USB Kabel oder Bluetooth kann der Computer dann mit den NXT Stein kommunizieren. Dadurch erreichen wir eine möglichst kleine Grundfläche, wodurch Drehungen auf kleinem Raum möglich sind. Das ist wichtig, weil der Roboter so durch schmalere Stellen des Parkours kommen kann und auch nicht viel Fläche einnimmt. In Abbildung 1 ist erkennbar, dass

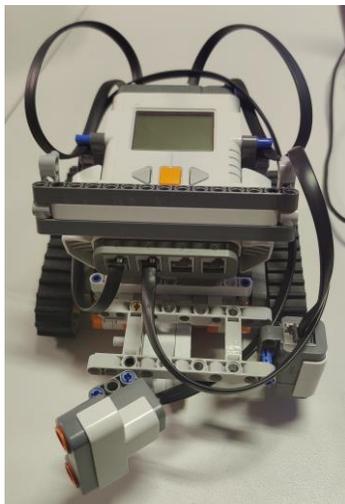


Abbildung 1: Roboter mit Smartphone-Halterung.

zusätzlich zu den Motoren noch zwei weitere Sensoren verwendet werden. Diese sind vor den Motoren und dem NXT Stein angebracht. Der Ultraschallsensor ist ganz vorne in einem 45 Grad Winkel angebracht. Dieser 45 Grad Winkel ermöglicht die Abstandsmessung nach vorne sowohl als auch nach rechts. Dadurch kann der Roboter den Abstand nach rechts halten und gleichzeitig davor geschützt werden mit der Wand, auf die er sich zubewegt zu kollidieren. Der Farbsensor ist genau vor der linken Kette platziert und nach unten gerichtet um später Farben zu erkennen. Die Kabel, die den NXT Stein mit den Motoren und Sensoren verbinden wurden so platziert, dass sie die Bewegung des Roboters nicht einschränken und die Messungen der Sensoren nicht beeinträchtigen oder verfälschen. Zusätzlich gibt es eine Halterung, in die ein Smartphone eingesetzt werden kann um die Fahrt aus Fahrersicht zu filmen.

B. Autonomes Fahren

Grundsätzlich wird das autonome Bewältigen des Parkours dadurch geschafft, dass der Roboter sich mit Hilfe des Ultraschallsensors an der rechten Wand hält. Dazu wird permanent der Abstand gemessen und ausgewertet. Dadurch wird gleichzeitig eine schräge Ausrichtung korrigiert. Schräge Ausrichtungen kommen dadurch zustande, dass die Motoren sich nicht exakt gleich verhalten und deswegen unterschiedlich schnell beschleunigen und abbremsen. Dadurch kommen ungenaue Drehungen zustande welche ausgeglichen werden müssen. Wie in Abbildung 2 erkennbar ist, sorgt die 45 Grad Ausrichtung des Ultraschallsensors an

der Front dafür, dass der Abstand zur rechten Wand und zur vorderen Wand gemessen werden kann. Wenn eine 90 Grad

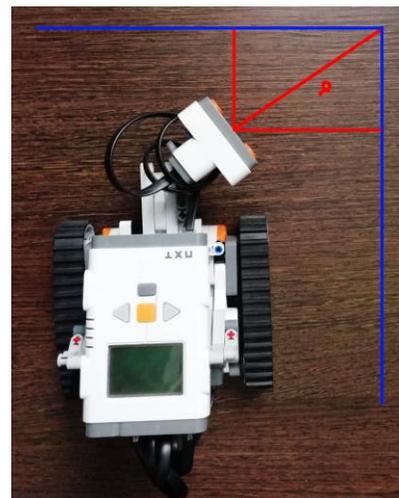


Abbildung 2: Abstandsmessung nach rechts und vorne durch 45 Grad Ausrichtung des Ultraschallsensors

Ecke nach links kommt kann der Roboter nun die Wand vor sich zu seiner rechten Wand machen indem er sich gegen den Uhrzeigersinn dreht. Danach orientiert er sich wieder an der von ihm rechten Wand. Dadurch kann Roboter nun Abbiegen ohne gegen eine Wand zu fahren. Durch die Ungenauigkeiten der Motoren muss der Roboter sich während der Fahrt ständig neu ausrichten. Dabei gibt es, wie in der Fuzzylogik beschrie-

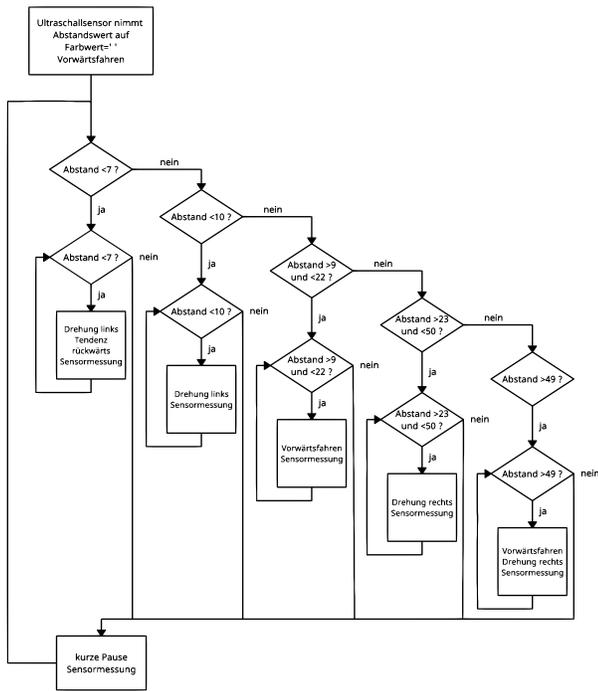


Abbildung 3: Programmablaufplan zum Algorithmus zur Abstandserkennung.

ben, 5 verschiedene Fälle, die durch die unterschiedlichen Abstandswertbereiche zustande kommen und dem entsprechend unterschiedliche Reaktionen hervorrufen damit der Roboter sich autonom durch den Parkour bewegen kann. Im ersten Fall ist der Abstand kleiner als 7. In diesem Fall dreht der Roboter sich gegen der Uhrzeigersinn und bewegt sich dabei leicht nach hinten. Dabei orientiert er sich immer noch an der rechten Wand. Durch diesen Falls wird verhindert, dass der Roboter gegen eine Wand fährt. In Fall zwei ist der gemessene Abstand zwischen 7 und 10. Dieser Fall tritt ein, wenn der Roboter bei der Fahrt etwas zu nah an die Wand heranfährt. In diesem Fall führt der Roboter eine Drehung auf der Stelle gegen den Uhrzeigersinn aus bis der optimale Abstand wieder erreicht ist. In dem Fall in dem der Roboter sich im optimalen Abstand zur Wand befindet ist der gemessene Abstand zwischen 9 und 22. Bei einem Abstand zwischen 9 und 22 steht der Roboter relativ parallel zur rechten Wand und soll geradeaus fahren indem beide Motoren mit gleicher Leistung nach vorne drehen. In Fall 4 ist der gemessene Abstand zwischen 23 und 50. Das entspricht einer leichten Auslenkung nach links und führt zu einer Drehung auf der Stelle im Uhrzeigersinn bis der Abstand wieder im optimalen Bereich ist. Im fünften Fall ist der gemessene Abstand größer als 49. Das tritt ein wenn eine Abbiegung nach rechts geschehen soll. Dafür muss der Roboter allerdings zuerst ein kleines Stück nach vorne fahren um nicht mit der Ecke zu verkanten. Danach Dreht er sich so lange im Uhrzeigersinn bis er wieder eine Wand findet, an der er sich orientieren kann. Dieser Algorithmus ist in Abbildung 3 beschrieben. Hier erkennt man auch, dass nach jeder Aktion der Ultraschallsensor wieder einen Abstand misst.

C. Farberkennung

Der Roboter soll nun neben dem Durchfahren des aufgestellten Parkours auch noch farbig gekennzeichnete Flächen erkennen und auf diese je nach Farbe reagieren. Hierbei soll zwischen roten Gefahrenflächen und grünen Opferflächen unterschieden werden. Wenn der Roboter eine grüne Opferfläche erkennt, soll er ein Signal in Form von einem Geräusch abgeben. Dieses Geräusch kann der NXT-Stein mit dem integrierten Lautsprecher erzeugen. Eine rote Fläche soll als Gefahrenfläche erkannt werden. In einer realen Anwendung wäre dies ein Loch, welchem der Roboter ausweichen muss. In Abbildung 4 ist der Programmablaufplan für die Farbmessung dargestellt. Der Algorithmus für die Abstandsmessung mit dem Ultraschallsensor wurde in den Algorithmus für die Farbmessung eingefügt, sodass der Roboter zusätzlich zur permanenten Abstandsmessung auch eine permanente Farbmessung durchführt. Wenn der Roboter eine Rote Farbfläche erkennt bleibt er zunächst stehen. Danach dreht er sich nach links, wobei er sich gleichzeitig etwas nach hinten bewegt. Das wird erreicht indem der linke Motor mit einer höheren Leistung nach hinten dreht als der rechte Motor sich nach vorne dreht. Dies wird so lange wiederholt bis er keine rote Farbe mehr erkennt. Danach orientiert er sich wieder an dem vom Ultraschallsensor gemessenen Abstand, was dazu führt, dass der Roboter sich nach vorne rechts bewegt weil der Abstand viel zu groß ist. Dadurch erkennt er wieder die rote Fläche und fängt wieder an sich nach links zu drehen. Dadurch bewegt der Roboter sich also durch ständige Rechtsdrehungen durch die Abstandsmessung und ständigen Linksdrehungen durch die Farbmessung langsam aber sicher an der roten Gefahrenfläche vorbei bis er wieder eine Wand vor sich erkennt an der er sich orientiert.

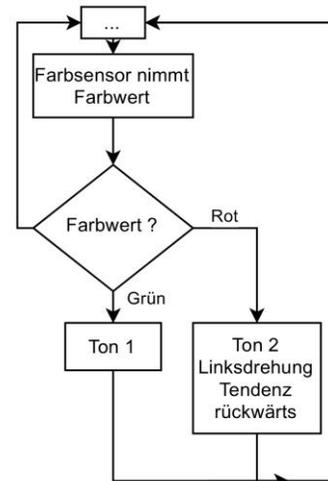


Abbildung 4: Programmablaufplan zum Algorithmus zur Farberkennung.

D. Videobegleitung

Wie im Aufbau erwähnt gibt es eine Halterung für ein Smartphone, welches die Fahrt aufnehmen soll. Das hat den Zweck, dass der Parkour im Nachhinein auch durch ein Video nachvollzogen werden kann. Das hat in einer realen Anwendung den Vorteil, dass ein darauf folgender Einsatz in diesem Gebiet besser geplant werden kann.

IV. ERGEBNISDISKUSSION

Nach dem zweiwöchigen Legopraktikum ist es uns gelungen ein Kettenfahrzeug aus Lego zu bauen welches durch Drehungen autonom einen Parkour in annehmbarer Zeit bewältigen kann. Zudem erkennt unser Aufklärungsroboter Gefahrenstellen, denen er ausweichen kann und Opferstellen an dem er ein akustisches Signal von sich gibt. Aufgrund der begrenzten Zeit ist es uns nicht gelungen eine automatische Kartenerstellung zu implementieren. Dahingegen haben wir aber eine Möglichkeit gefunden den Ablauf per Video festzuhalten, sodass die Fahrt später nachvollzogen werden kann. Während dem Arbeiten und Testen an dem Roboter sind uns allerdings auch einige Probleme aufgetreten. Durch die Verbindung mit Bluetooth waren wir zwar nicht mehr auf einen engen Raum gebunden, haben allerdings durch die Latenz immer wieder Ungenauigkeiten bei Drehungen erkannt. Dadurch mussten wir wieder auf Kabelverbindungen umsteigen. Nicht nur der Roboter, sondern auch der Parkour hatte Probleme. An 90-Grad Ecken, an denen der Roboter sich nach links drehen soll blieb er öfters hängen. Das ließ sich beheben indem der Winkel etwas vergrößert wurde. Da die Sensoren nicht dafür gedacht sind zu jeder Zeit sehr genaue Wert zu liefern und im Endeffekt auch nur Spielzeug sind, traten manchmal unerklärliche Werte auf, die den Roboter etwas durcheinander brachten. Das konnten wir beheben indem wir den Ultraschallsensor nicht horizontal sondern vertikal verbauten. Das größte Problem waren jedoch die Ungenauigkeiten in den Motoren. Öfters drehten sich die Motoren weiter als sie sollten bei gleichen Befehlen. Dadurch musste durch die Sensoren immer die Lage ausgeglichen werden, was ein Orientieren im Parkour nicht möglich gemacht hat. Somit mussten wir schließlich auch das Erstellen der Karte verzichten.

V. ZUSAMMENFASSUNG UND FAZIT

Wir haben von Anfang an geplant einen Aufklärungsroboter zu entwickeln, der durch einen Parkour fahren kann ohne dabei die Wände zu berühren und zudem auch noch verschiedene Stellen erkennt und schließlich eine Karte erstellt. Hierzu haben wir uns reale Aufklärungsroboter als Beispiel genommen und uns an ihnen orientiert. Wir haben es geschafft ein Kettenfahrzeug zu bauen, welches autonom durch einen unbekanntes Parkour fahren konnte und dies durch Videoaufnahme aufzeichnen konnte. Dabei hat das Fahrzeug Farben erkannt und hat dem entsprechend Töne an den dafür gedachten Stellen gemacht. Damit haben wir er geschafft einen Roboter zu bauen, der an Stelle von Menschen ein ungekanntes potenziell gefährliches Gebiet erforscht um somit Rettungsmissionen zu ermöglichen, ohne, dass sich dafür ein Mensch vorher in Gefahr begeben musste. Es ist zwar immer noch eine stark vereinfachte Variante von echten Aufklärungsrobotern, die mehr Herausforderungen haben als nur gerade Wände und farblich gekennzeichnete Flächen auf dem Boden. Es bestehen also noch sehr viele Verbesserungsmöglichkeiten. Mit mehr Ultraschallsensoren wäre es sicherlich möglich gewesen die Umge-

bung besser zu erfassen und den Roboter besser durch den Parkour zu manövrieren.

LITERATURVERZEICHNIS

- [1] Alexander Behrens (2021). RWTH - Mindstorms NXT Toolbox (<https://www.mathworks.com/matlabcentral/fileexchange/18646-rwth-mindstorms-nxt-toolbox>), MATLAB Central File Exchange. Retrieved March 28, 2021.
- [2] Wikipedia, The Free Encyclopedia – FuzzyLogik. <https://de.wikipedia.org/wiki/Fuzzylogik> - 28.03.2021