

Farberkennungsgesteuerte Ballschussmaschine

Fabian Meyer, Elektro- und Informationstechnik
Otto-von-Guericke-Universität Magdeburg

Zusammenfassung—Auch in diesem Jahr wurden an der Otto-von-Guericke-Universität Magdeburg Studenten des ersten Semesters des Studiengangs Elektro- und Informationstechnik im Projektseminar LEGO Mindstorms damit betraut, einen Roboter zu konstruieren und in MATLAB zu programmieren, der selbständig eine Aufgabe bewältigen soll. In diesem Paper wird die Entwicklung, Konstruktion und Programmierung eines Roboters beschrieben, welcher in der Lage ist, durch Zielerkennung farblich markierte Objekte im Raum zu erkennen, diese anzusteuern und letztendlich auch mit einer Ballschussmaschine zu beschießen.

Schlagwörter—Ballschussanlage, Fahrzeug, LEGO Mindstorms, Roboter, Zielerkennung

I. EINLEITUNG

DIE Erkennung von Objekten durch Computer, ob von markierten oder unmarkierten spielt eine große Rolle in vielen Bereichen unserer modernen Welt. Ob in medial sehr wirksamen, und allgemein stark umstrittenen Bereichen wie der militären Anwendung, oder in, dem Ottonormalverbraucher eher verschlossenen Gebieten, wie der zivilen Logistik, solche Erkennungssysteme spielen vielerorts eine wichtige Rolle.

II. VORBETRACHTUNGEN

A. LEGO Mindstorms

Erst einmal sollten die zur Verfügung stehenden Bauteile ein wenig genauer angesehen werden. Das LEGO Mindstorms Education Set besteht aus dem NXT-Brick, also dem zentralen Baustein des Roboters, welcher vier Sensoren-Eingänge und drei Motoren-Ausgänge zur Verwendung zur Verfügung stellt. Zum Anschließen sind fünf verschiedene Sensoren verfügbar: Ein einfacher, binärer Tastsensor, ein Geräuschsensor, ein Farbsensor, ein Lichtsensor und ein Ultraschallsensor [1]. Daraus, bis zu drei interaktiven Servomotoren und normalen LEGO-Technik-Bauteilen sollte ein Roboter konstruiert werden.

B. MATLAB

Für die Programmierung des Roboters wurde MATLAB verwendet, eine Software zur Lösung von mathematischen und ingenieurtechnischen Problemstellungen. Entwickelt wird MATLAB von MathWorks, welches laut eigener Angabe „der führende Entwickler von Software für mathematische Berechnungen für Ingenieure und Wissenschaftler“ [2] sei. Für die Kommunikation zwischen MATLAB und dem NXT wird die dazu bestimmte Software des Lehrstuhls für Bildverarbeitung der RWTH Aachen verwendet [3].

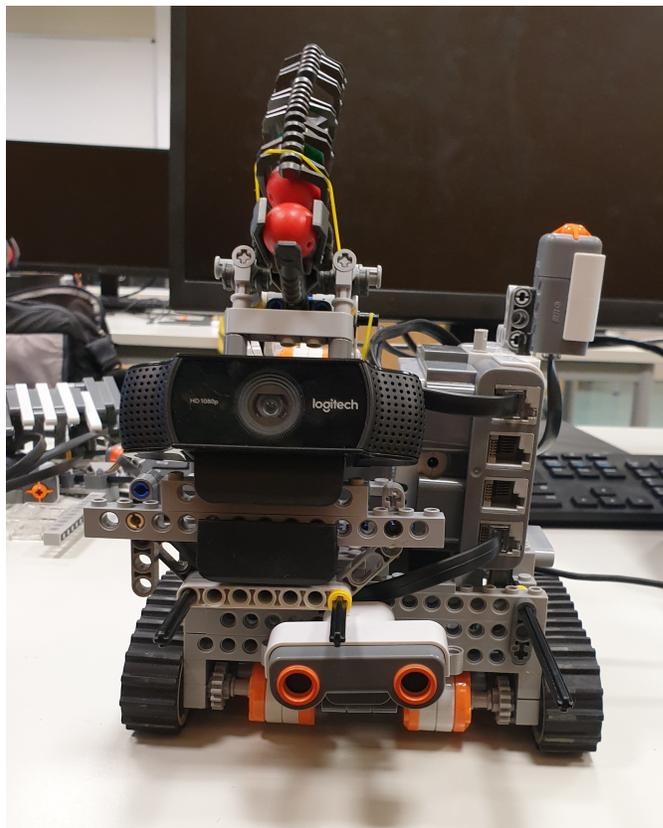


Abbildung 1. Frontansicht des Roboters

III. KONSTRUKTION UND PROGRAMMIERUNG

A. Zwei Funktionsmodi als Grundidee

Die Idee zu den grundlegenden Funktionsmodi des Roboters war von Beginn an, dass der Roboter über einen autonomen und einen manuellen Modus verfügen sollte. Dabei war das Problem zu überwinden, dass der LEGO-NXT-Baustein nur 4 Sensor-Eingänge hat, welche bei der erdachten Fernsteuerungs-Architektur bereits alle belegt wären, und uns somit keine für den autonomen Modus mehr übrig wären. Dazu wurde ein System entwickelt, wobei die Kabel der Fernbedienung für den manuellen Modus erst angesteckt werden müssten, um diesen zu benutzen. Dies wurde so realisiert, dass bei Programmstart entweder ein bestimmter und bereits initialisierter Tastsensor der Fernbedienung für den manuellen Modus, oder ein bereits am Roboter fest verbauter Sensor, welcher den autonomen Modus startet gedrückt werden müssen. Der für den autonomen Modus benötigte Ultraschall-Sensor und der zusätzliche Tastsensor am Roboter sind bei Initialisierung bereits verbaut und verbunden, und letzterer muss ausgelöst

werden, um den autonomen Modus zu starten. Dies ist auch im Programmablaufplan in Abbildung 6 dargestellt.

B. Autonomer Modus

Im autonomen Modus macht der Roboter immer wieder Bilder mit seiner Webcam, welche über USB an einen Computer angeschlossen ist und analysiert diese dann auf rote Objekte. Findet er keins, dreht er sich ein Stück weg und fährt vorwärts, wobei er auch fortwährend Bilder aufnimmt und bei Objektfund sofort anhält. Stößt der Roboter auf eine Wand, welche er mit seinem Ultraschallsensor erkennt, fährt er ein Stück zurück, dreht sich von ihr weg und setzt seine Fahrt fort. Sobald er ein Ziel in der Kamera gefunden hat, hält er sofort an und richtet sich durch feine Motoransteuerungen darauf aus. Ist er ausgerichtet, passt er seine Distanz zu Ziel mit Hilfe des Ultraschallsensors an. Sobald dies erfolgt ist, schießt er auf das Ziel und nimmt danach ein Bild auf. Ist das Ziel nicht mehr zu sehen, also umgeklappt, dreht der Roboter sich zum nächsten Ziel. Sollte das Ziel nicht von der Bildfläche verschwinden, also nicht getroffen worden sein, was bei der materialbedingten Inkonzanz der Schüsse passieren kann, wird er solange schießen, bis das Ziel getroffen wurde.

C. Wahl des Antriebssystems

Die Wahl des Antriebssystems fiel sehr schnell auf einen auf Ketten basierten Antrieb, da er es dem Roboter ermöglicht, auch unebenes Terrain mit Leichtigkeit zu überwinden und sich auf der Stelle zu drehen. Zusätzlich besteht bei einem eher kleineren Fahrzeug wie diesem die Möglichkeit Allrad-Kettenantriebe, also Kettenantriebe bei denen alle Räder Antriebsräder sind und keine Laufräder vorkommen, zu verwenden, da die auftretende Reibung eher zu vernachlässigen ist. Dies verleiht dem Roboter die Möglichkeit sich auch mit nicht montierten oder während des Betriebs abgefallenen Rädern problemlos fortzubewegen. Eine Seite des Allrad-Getriebes ist in Abbildung 2 dargestellt.



Abbildung 2. Ansicht einer Seite des Allrad-Ketten-Getriebes

D. Funktion der Webcam

Die Farberkennung mit der Webcam funktioniert so, dass die Webcam ein Bild aufnimmt und nur die Rotwerte des RGB-Bildes als Matrix ausgibt. Dabei entsteht eine Matrix vom Format der Auflösung der Webcam, welche aus den Rotwerten von 0 bis 255 besteht. Dies ist beispielhaft in Abbildung 3 dargestellt. Das Bild wird nun erst gefiltert um unnötiges Rauschen zu vermeiden. Dabei erhält jeder Pixel im

$$A = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 255 & 255 & 255 & 255 & 0 \\ 0 & 255 & 255 & 255 & 255 & 0 \\ 0 & 255 & 255 & 255 & 255 & 0 \\ 0 & 255 & 255 & 255 & 255 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

Abbildung 3. Beispielhafte Darstellung eines perfekt roten Quadrates, aufgenommen in 6 x 6-Auflösung

gefilterten Bild den durchschnittlichen Wert der neun Pixel im 3 x 3-Bereich um ihn herum und sich selbst im ursprünglichen Bild. Daraufhin wird nach Farbwerten gefiltert. Dabei wird jeder Wert, der das 0.18-fache der maximalen Licht-/Farbstärke überschreitet zu einer Eins, und die, die das nicht tun zu einer Null. Dies ist in Abbildung 4 veranschaulicht, außerdem ist der Unterschied zwischen Abbildung 3 und Abbildung 4 zu betrachten. Vor dem eigentlichen Kenntlichmachen der

$$B = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

Abbildung 4. Umwandlung auf Nullen und Einsen

roten Objekte werden noch Objekte, welche aus weniger als 300 Pixeln bestehen entfernt, damit vom Computer nicht kleinste rote Objekte fälschlicherweise als Ziele erkannt werden. Danach kann mit dem Markieren begonnen werden. Dafür wird um das rote Objekt eine sog. „Bounding-Box“ erzeugt, also einfach ein Kasten um das Objekt. Im Zentrum dieser „Bounding-Box“ wird daraufhin ein sog. „Centroid“ markiert, ein Objekt, welches in der Mitte des roten Objektes liegt. Dies ist in Abbildung 5 veranschaulicht. Die Koordinaten des „Centroids“ in Kamerabild verwendet der Roboter daraufhin, um sich auf das Ziel auszurichten.



Abbildung 5. Eine „Bounding-Box“ und ein „Centroid“ um ein rotes Objekt

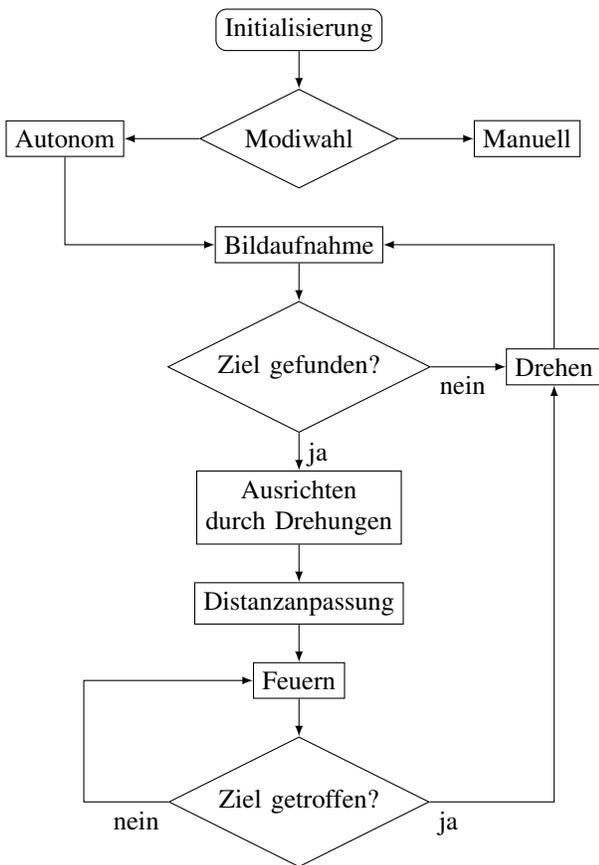


Abbildung 6. Programmablaufplan des Roboterprogrammes

E. Die Ballschussanlage

Die Ballschussanlage stellte eine größere Herausforderung dar als eigentlich gedacht. Beim Herausdrücken der Bälle aus der Abschussvorrichtung musste nämlich mehr Kraft aufgewandt werden, als zuerst angenommen. Sobald die Vorrichtung konstruiert war, stellt sich ein zweites Problem dar: Durch die große Kraft, die durch den Motor aufgewandt werden musste, verbog sich das Plastikgestell so sehr, dass kein Abschuss zu Stande kam. Dies wurde mit Gummibändern behoben, welche so gespannt wurden, dass sie die Biegungen so weit einschränkten, dass Abschüsse möglich waren. Eine Seitenansicht der fertigen Ballschussanlage ist in Abbildung 7 zu sehen.

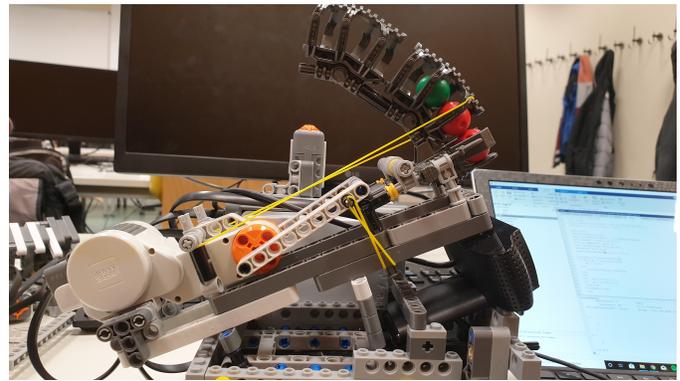


Abbildung 7. Seitenansicht der Ballschussanlage

B. Ungenauigkeit der Sensoren

Die Sensoren lassen bei manchen Aspekten der Genauigkeit etwas zu wünschen übrig. Zum Beispiel muss der Ultraschallsensor genau senkrecht positioniert werden um die Entfernung zuverlässig und akkurat auslesen zu können. Sobald eine Rundung oder eine Ecke anvisiert wird, verliert Entfernungsmessung an Zuverlässigkeit.

V. ANDERE VERSIONEN UND VERWORFENE IDEEN

A. Position der Kamera

In älteren Versionen des Roboters fand sich die Webcam noch neben der Ballschussmaschine. Dadurch entstand jedoch eine Verschiebung der Schusslinie, da sich der Roboter mit Hilfe der Kamera ausrichtet, jedoch, wenn die Kamera zentriert ist, immer an Ziel vorbei schoss. Deshalb befindet sich die Kamera in der finalen Version direkt unter der Kamera. So richtet der Roboter sich perfekt aus.

B. Höhenabhängige Distanz

Außerdem war eine Funktion geplant, bei welcher der Roboter aus der Entfernung zu Ziel und dem Winkel, in dem es sich in der Kamera befindet, die Höhe des Ziels berechnet, und so entsprechend seine Distanz dynamisch anpasst. Die Unzuverlässigkeit des Ultraschallsensors zwang uns jedoch dazu, diese Idee zu verwerfen.

C. Lenkung durch Zu- und Abkopplung der Ketten

Zusätzlich stand die Idee im Raum, die Lenkung des Roboters so zu realisieren, dass mit einem kupplungsähnlichen LEGO-Technik-Teil eine Seite der Ketten abgeschaltet werden könnte. So hätte man zum Treiben der Ketten nur noch einen Motor benötigt. Allerdings hätten dann für die Schaltung der Kupplung ein Motor verwendet werden müssen und Drehmoment der Ketten wäre verloren gegangen, da sie nur noch mit einem Motor betrieben werden würden. Deshalb fiel Entscheidung auf die simple Konstruktion mit einem Motor pro Seite. Außerdem kann so beim Lenken eine Seite vorwärts und die andere rückwärts drehen, anstatt eine Seite im Leerlauf zu haben. Dies verkleinert den Wendekreis noch weiter, nämlich auf die Länge des Fahrzeugs.

IV. PROBLEMSTELLUNGEN

A. Limitationen des NXT

Der NXT, also das zentrale Steuer- und Empfangsmodul für die Motoren und Sensoren bietet eine Limitationen. Dazu gehören die begrenzte Anzahl an Sensor-Eingängen, nämlich vier Stück. Die Anzahl der Motor-Ausgänge ist mit drei Stück noch stärker begrenzt. Dies schränkte einige Ideen, die entwickelt wurden stark ein. Zusätzlich streikt der NXT dann, wenn er zu viele Befehle zu schnell bekommt, was beim wiederholten Abfragen von einigen Sensoren sehr schnell passiert.

VI. ERGEBNISDISKUSSION

Am Ende dieses Projektes steht ein Roboter, dargestellt in Abbildung 1, welcher zuverlässig rot markierte Objekte anvisiert und trifft. Auch das Bewegen im Raum funktioniert bei richtigen Bedingungen zuverlässig und gut. Diese richtigen Bedingungen sind bei der Bewegung im Raum relativ weit gefasst, wobei ein komplexerer Aufbau, wie zum Beispiel eine Lücke in der aufgebauten Wand an falscher Stelle ausreicht, um den Ultraschallsensor in die Irre zu führen. Die Funktion des Anvisierens ist allerdings einfacher zu stören. Trägt jemand im Raum ein rotes T-shirt wird derjenige anstatt des eigentlichen Ziels vom Roboter beschossen.

VII. ZUSAMMENFASSUNG UND FAZIT

Wir haben also einen Roboter entworfen, welcher zuverlässig markierte Objekte im Raum findet, ansteuert und beschießt. Die Realisierung dessen mit einer Webcam lässt Platz für relativ einfache und schnell Änderungen. So könnte zum Beispiel mit einer einfachen Änderung im Code die gesuchte Farbe geändert, oder mit der Einbettung von Computer-Vision-Bibliotheken komplexe Bilder erkannt werden, um die Nützlichkeit des Roboters in immer neuen Anwendungsgebieten zu gewährleisten. Auch könnten bei neueren Versionen des LEGO-Mindstorms-Bausatzes neue Funktionen des Roboters möglich werden. Bei neueren Sensoren könnte auf komplett neue Größen der realen Welt reagiert werden. Ebenfalls könnten bei hinzugefügten zusätzlichen Motor-Ausgängen weitere Funktionen hinzukommen, wie zum Beispiel das Heben und Senken der Abschussvorrichtung, oder ein Arm, welcher die getroffenen Ziele selbst wieder aufstellt. Der Vielfalt an möglichen zukünftigen Entwicklungsschritten und Versionen ist also nahezu keine Grenze gesetzt.

LITERATURVERZEICHNIS

- [1] BASTGEN, PETER, http://www.bastgen.de/schule/Projektkurs/robotik%209/NXT%20User%20Guide/assets/languages/german/print_all/content/9797_LME_UserGuide.pdf, 05.03.2020
- [2] MATHWORKS, <https://de.mathworks.com/>, 05.03.2020
- [3] RWTH AACHEN UNIVERSITY, LEHRSTUHL FÜR BILD-VERARBEITUNG, <https://www.mindstorms.rwth-aachen.de/trac>, 05.03.2020