

# LEGO Mindstorms Ballschussmaschine

Jonah Adrian Walther, Elektro- und Informationstechnik  
Otto-von-Guericke-Universität Magdeburg

**Zusammenfassung**—Im folgenden Paper geht es um die Planung, Konstruktion und Programmierung eines autonomen Ballschussroboters der mittels Farb- und Distanzerkennung auf farbige Ziele schießen kann. In allen Bereichen sind einige Probleme aufgetreten die ebenfalls erläutert werden und zum Großteil behoben wurden, weiterhin werden die Anwendungsbereiche aufgezeigt.

**Schlagwörter**—Ballschussmaschine, Distanzerkennung, Farberkennung, LEGO Mindstorm, Otto-von-Guericke-Universität, Projektseminar, Roboter

## I. EINLEITUNG

WIE auch schon in vergangenen Jahren ist auch im Jahre 2020 wieder das LEGO Mindstorms Projektseminar der Otto-von-Guericke-Universität abgehalten worden, um den Studenten einen Einblick in die Welt der Robotik zu verschaffen. In dem zweiwöchigem Seminar sind viele Studenten in Gruppen mit der Programmiersprache MATLAB vertraut gemacht worden, um dann mit dem LEGO Mindstorms Set einen Roboter zu erschaffen, der eine Aufgabe bewältigen kann. Heutzutage gehören Roboter auf unserer Welt fast überall zum Leben dazu. Sie helfen uns alltägliche Probleme zu überstehen, helfen uns bei der Arbeit und fordern uns heraus unsere Leistungen zu verbessern. Roboter sind schneller, präziser und können Orte erreichen, die für den Menschen unerreichbar scheinen. Die Entwicklung ist noch lange nicht am Ende, vor allem in der Industrie ist der Roboter gern gesehen, da jeder Schritt genauestens programmiert werden kann, er schwere Objekte mit Leichtigkeit bewegen lässt und immer auf den Punkt genau arbeitet, nimmt er hier eine der größten Rollen ein. Aber auch im Sport hat er sich hervorgetan, beim Trainieren von Muskeln und zum Kraftaufbau wird er gerne eingesetzt. Um die Koordination, Schnelligkeit und Reaktionsgeschwindigkeit zu verbessern nutzen viele Sportler im Rückschlagsport (z. B. Tennis) sogenannte "Ballschussanlagen", das sind Roboter die verschiedenste Arten von Bällen abschießen können. Sie simulieren dem Sportler einen Gegenspieler, wenn keiner vorhanden ist. Im militärischen Bereich haben sich Drohnen weit nach vorn gearbeitet, das sind führerlose Fahrzeuge, welche zu Land, zu Wasser oder in der Luft agieren können, um zum Beispiel ein Gelände zu Kartografieren. Eine Kamera zeichnet Bilder der Umgebung auf und eine künstliche Intelligenz erarbeitet daraus eine Karte, die für den Menschen anschaulich ist. Unser Ballschussroboter verbindet den militärisch und sportlich genannten Bereich und bietet einem ein autonomes Fahrzeug welches mittels einer Kamera die Umgebung erfasst und Bälle auf Ziele schießt

## II. VORBETRACHTUNGEN

Um die Funktionen des Roboters besser erklären zu können, müssen zuerst die Funktionen des Lego Mindstorm-NXT-2.0 dem Kernelement des Systems erläutert werden, sowie die extra implementierte Webcam.

### A. NXT

Der NXT-Legostein ist eine programmierbare Steuereinheit die 4 Eingänge für Sensoren und 3 Ausgänge für Motoren unterstützt. Bei unserem Roboter wurde der Ultraschallsensor und der Tastsensor verbaut. Der Tastsensor befindet sich in Abbildung 1 ganz links und funktioniert nach dem gleichen Prinzip wie ein herkömmlicher Taster, solange er gedrückt wird, gibt er ein Signal aus.



Abbildung 1. NXT 2.0 <https://jaxenter.de/lego-mindstorms-nxt-2662>

### B. Ultraschallsensor

Der Ultraschallsensor des LEGO-Mindstroms-2.0-Set funktioniert nach einem Echo-Prinzip. Wie in Abbildung 2 zu sehen, wird von einem Sender ein Schallimpuls ausgesendet der an Objekten reflektiert wird und vom Empfänger ausgelesen werden kann. Die Zeit welche zwischen dem Senden und Empfangen vergeht, dient dabei als Referenz für die Distanzberechnung. Der Ultraschallsensor des LEGO-Mindstroms-2.0-Sets kann Objekte die bis zu 2,5 m entfernt sind bestimmen, die Distanz gibt dieser dabei in cm aus.

### C. Webcam

Die Implementierte Webcam ist eine Logitech C920 HD Webcam mit einer Bildauflösung von 1920\*1080 Pixeln, welche über einen USB Port angeschlossen werden kann. Zusätzlich ist die Linse im Winkel verstellbar.

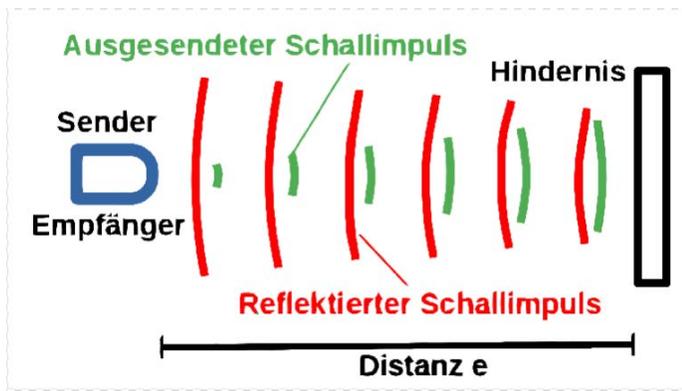


Abbildung 2. Funktion des Ultraschallsensor <https://www.technik-unterrichten.de/Robotik/Ultraschallsensor/Ultraschallsensor.php>



Abbildung 3. Logitech Webcam C920 HD <https://www.amazon.de/Logitech-C920-HD-Pro-Webcam-Videogespräche-Videoaufnahmen-Full-HD-Stereo-Mikrofonen/dp/B006A2Q81M>

### III. HAUPTTEIL

#### A. Mechanische Konstruktion

Für die Konstruktion standen den Gruppen mehrere Lego-Mindstorms-Baukastensets zur Verfügung, außerdem standen mehrere Sensoren zur freien Verfügung. Mit dem Baukasten wurde zuerst eine stabile Grundkonstruktion errichtet, welche das Gewicht des NXT-Legosteins standhält. An diese Grundkonstruktion wurde ein Antrieb an der Unterseite angebracht, wofür zwei Motoren verwendet wurden. Die Motoren treiben jeweils eine Zahnradreihe an, die für einen Kettenantrieb verwendet werden. Oberhalb der Grundkonstruktion wurde der NXT-Legostein befestigt und eine Abschussvorrichtung für Legobälle angebracht, die Abschussvorrichtung wurde zusätzlich mit einem Motor versehen. An der Vorderseite ist der Ultraschallsensor angebracht worden und an der rechten

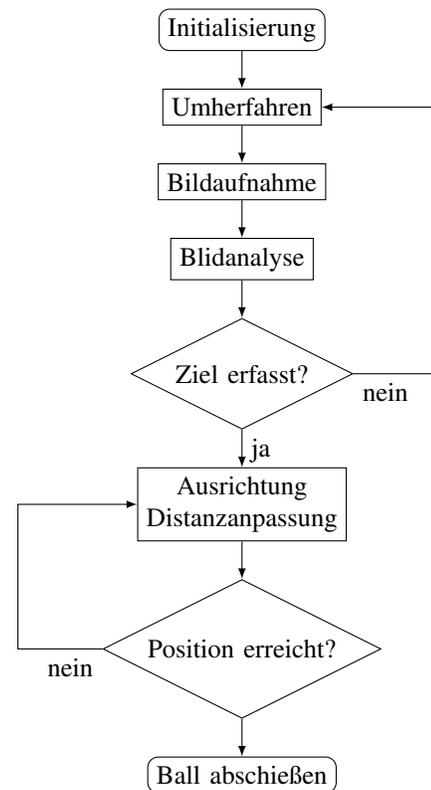


Abbildung 4. Programmablaufplan zur Erklärung des MatLab Program

Seite der Tastsensor. Oberhalb des Ultraschallsensors wurde eine kleine Halterung zum Fixieren der Webcam positioniert, sowie die Webcam selbst.

#### B. Programmlauf

Anhand des Programmablaufplans ist eine grobe Angabe des Programms in MATLAB und dient zur Visualisierung für Personen, die sich mit der Programmiersprache nicht auseinandergesetzt haben.

#### C. Bildaufnahme und Analyse

Wie in Abbildung 4 gezeigt wird ein Bild von der Webcam aufgenommen und dann analysiert. Das von der Webcam aufgenommene Bild wird in MATLAB in einer Matrix gespeichert, welche die Auflösung der Kamera als Größe festlegt (1920\*1080 also 2 073 600 Stellen in der Matrix). Jeder aufgenommene Pixel, wird dabei in einem Array gespeichert, welches dann im Matrix-Element implementiert wird. Der Pixel im Array ist als RGB-Code gespeichert, der RGB-Code steht für die Grundfarben Rot, Grün und Blau im RGB-Farbraum. Der RGB-Farbraum lässt sich wie einen Würfel beschreiben in dem alle möglichen Farben in einen drei-dimensionalen kartesischen Koordinatensystem von 0 bis 255 aufgelistet sind. Die Koordinaten bestimmen dabei dem Rot-, Grün- und Blauanteile. Die Matrix in MATLAB wird im nächsten Schritt untersucht, jedes Element der gespeicherten Arrays wird auf bestimmte Werte geprüft die in dem Farbwürfel im roten Bereich liegen. Der darauf folgende Schritt ändert jedes Array, welche den roten Bereich nicht erreichen zu einem Array voller

Nullen und alle Arrays die den roten Bereich erreichen zu einem Array voller Einsen. Alle Arrays werden danach in den Durchschnittswert ihres Inhaltes umgeschrieben und in einer neuen Matrix gespeichert. Diese neue Matrix wird im nächsten schritt auf Nullen und Einsen untersucht und umrahmt alle Gruppen von Einsen mit einem Kasten. Innerhalb des Kastens kann ein Zentrierungspunkt gesetzt werden, um den Mittelpunkt des roten Objektes zu definieren, anhand des Mittelpunktes richtet sich der Roboter dann aus Abbildung 5



Abbildung 5. Bildhafte Umsetzung der Matrixanalyse

#### D. Umherfahren

Außerdem wird in Abbildung 4 gezeigt, dass sich das Programm in einer Schleife befindet, welche so lange abläuft wie kein Ziel erkannt wurde. Dabei fährt der Roboter gerade aus und untersucht mit dem Ultraschallsensor den Abstand zu einem Objekt vor ihm, zum Beispiel einer Wand. Währenddessen erzeugt er jede Sekunde ein Bild mit der Webcam zum Analysieren und solange auf diesem Bild kein Zielobjekt zu erkennen ist, fährt er weiter gerade aus. Sollte er kein Ziel erkennen und sich einem Hindernis auf unter 20 cm Distanz befinden, stoppt der Roboter und dreht sich um ca. 90° zur Seite, danach macht er mit seinem Ablauf weiter.

#### E. Ausrichtung und Distanzanpassung

Zusätzlich zeigt die Abbildung 4 eine zweite Schleife welche sich um die Position des Roboters zu seinem Ziel kümmert. Dabei nutzt er den Mittelpunkt des roten Objektes in seiner Bild Matrix, um die horizontale Ausrichtung zu bestimmen. Da sich das Kamerabild direkt proportional zur Matrixgröße verhält, dreht sich der Roboter mittels Antreiben des Kettenantriebes in unterschiedliche Richtungen auf der Stelle und das Objekt in der Mitte des Bildes zu positionieren. Danach ermittelt der Roboter mittels dem Ultraschallsensor den Abstand zu seinem Ziel und passt die Entfernung auf seine Schussreichweite an.

### IV. ERGEBNISDISKUSSION

Während des Baus und der Programmierung sind viele Probleme aufgetreten und es gibt auch einige die nicht gelöst werden konnten. Zuerst hat die mechanische Konstruktion große Fehler verursacht, dabei hat sich die Ballabschussvorrichtung durch den enormen Druck, der aufgewendet werden musste,

um den Ball abzufeuern sehr instabil angefühlt und sich oft mal selbst auseinander genommen hat. Gelöst wurde die fehlerhafte Abschussvorrichtung dann mit Gummibändern als Fixierung, was den Großteil des Problems behoben hat. Leider entstand dabei sehr viel Stress aus mechanischer Sicht auf die Bauteile, was diese wohl nicht auf ewig überstehen werden. Das nächste Problem welches aufgetreten ist, war die Position des Roboters zu seinem Ziel. Da sich die Ausrichtung zur Mitte der Matrix bewegt, muss ein bestimmter Bereich festgelegt werden, da die Motoren nicht präzise genug arbeiten, dabei haben wir den Bereich vorerst zu klein gewählt, wodurch sich der Roboter immer um den Mittelpunkt rumgedreht hat und nie zur Ausrichtung der Distanz gekommen ist. Ein Problem welches nicht gelöst werden konnte ist, dass der Abschusswinkel konstant ist und somit die Objekte zum Abschießen auf derselben Höhe aufgestellt werden müssen da sich eine variable Höhe aus programmier und technischer Sicht aktuell nicht lösen ließ. Mit einem weiteren Motor wäre dies möglich gewesen allerdings ist keinen Platz mehr für einen weiteren Motor vorhanden, weshalb dieses Lösung leider weg fiel. In programmtechnischer Hinsicht war das Programm zu Beginn recht unordentlich, bestimmte Bereiche waren überflüssig und Fehler haben viele Probleme gemacht. Dieses Problem wurde durch Unterprogramme gelöst, welche für sich ordentlich und fehlerfrei sind und dann nur noch in einem Hauptprogramm zusammengeführt werden. Das letzte Problem, welches aufgetreten war, ist wenn mehrere Objekte auf dem Bild sind. Der Roboter hat dann das Objekt, welches am weitesten links ist fokussiert und sich ausgerichtet aber dann abgebrochen da sich ja noch eines auf der rechten Seite gefunden hat. Durch eine simple Programmanweisung wurde ein Objekt favorisiert und zuerst angesteuert.

#### A. Ergebnis

Am Ende des Projektes ist ein Roboter vorgestellt worden, welcher sich von allein in einem begrenzten Raum fortbewegt und auf rote Ziele feuern kann.

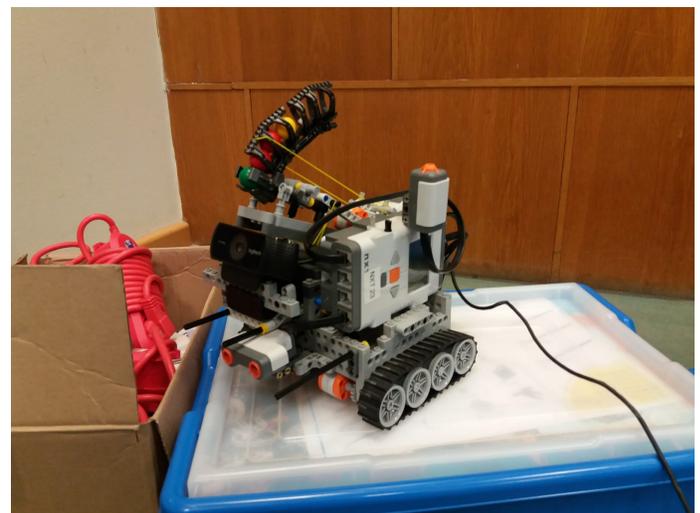


Abbildung 6. Finale Konstruktion

## V. ZUSAMMENFASSUNG UND FAZIT

Zusammengefasst wurde ein funktionstüchtiger Lego-Roboter gebaut, welcher die Aufgabe farbige Ziele abzuschließen, erfolgreich durchführen kann. Dabei kann er allein in einem begrenzten Bereich erkunden und von allein die Ziele ausfindig machen. Sich nach den Zielen zu orientieren und zu beschließen erfolgt über die Webcam und die Abschussvorrichtung. In Zukunft können noch weitere Verbesserungen vorgenommen werden, welche zum einen die Präzision verbessern könnten. Mittels eines Algorithmus soll der begrenzte Bereich nach einem Muster abgefahren werden und nicht mehr dem Zufall überlassen sein. Außerdem wäre eine Umstrukturierung der Abschussvorrichtung notwendig, um die Bauteile zu entlasten und eine allgemeine Halterung die jede Art von Webcam halten und nicht ausschließlich die aus Abbildung 3.

## ANHANG

Hier ist ein Video welches zum Verständnis der Bildanalyse:  
<https://www.youtube.com/watch?v=W9gsIOgFFo0>

## VI. LITERATURVERZEICHNIS

[1] Wikipedia, RGB-Fabraum,  
<https://de.wikipedia.org/wiki/RGB-Fabraum>