

Manipulator

Vladyslav Karkishko, Elektrotechnik und Informationstechnik
Otto-von-Guericke-Universität Magdeburg

Abstract — Diese Forschungsarbeit zielt darauf ab, einen Roboter-Manipulator zu entwickeln, zu bauen und zu programmieren, der mit dem Steuerungsblock LEGO Mindstorms EV3 interagieren kann. Die Hauptaufgabe besteht darin, ein System zu schaffen, das in der Lage ist, Objekte mithilfe eines Sensors zu erkennen und sie mithilfe von Motoren zu bewegen. Der Softwareteil des Projekts wurde unter Verwendung der Programmiersprache MATLAB entwickelt.

Schlagwörter — EV3, LEGO Mindstorms, mechanische Krallen, Roboter-Manipulator

I. EINLEITUNG

Ein beladener und bewegter Roboter-Manipulator (Abbildung 1) stellt eine Lösung für eine Vielzahl von Anwendungen dar. Seine Funktionalität kann in der Logistik, der Fertigung, der Lagerhaltung und anderen Bereichen, in denen Automatisierung und effizienter Güterumschlag von größter Bedeutung sind, gefordert werden. Die Hauptidee war das Heben und Bewegen von Lasten. Es wurde jedoch Wert darauf gelegt, die Vielseitigkeit des Roboters zu erhöhen. Das war das Hauptproblem, denn die LEGO-Teile erlaubten es nicht, vielseitige Lasten zu heben, also war es das Ziel, über LEGO hinauszugehen. Durch die Entwicklung von Modifikationen und Verbesserungen wurde das Spektrum der Aufgaben, die erfolgreich von unserem Roboter ausgeführt werden können, erheblich erweitert.



Abbildung 1: Roboter-Manipulator

II. VORBETRACHTUNGEN

Der gesamte Mechanismus der mechanischen Krallen basiert auf Sechsecken, so dass die Last mit Hilfe eines mittelgroßen Motors angehoben und abgesenkt wird. Der Roboter hat mit nur einem Sensor zur Objekterkennung, dem Touchsensor (Taste) korrekt funktioniert. Insgesamt werden drei Motoren und zwei Sensoren verwendet. Der mittlere Motor ist für das Aufnehmen und Laden des Objekts zuständig, während die beiden größeren Motoren für die Bewegung des Roboters verantwortlich sind.

A. Grundprinzip des mechanischen Krallenmechanismus

Die Idee, mechanische Krallen (Abbildung 2) zu bauen, beschränkte sich auf LEGO-Teile. In diesem Zusammenhang tauchte die Frage auf, an welchem Teil der Antriebseinheit sie befestigt werden sollten und welche Art von Last der Manipulator heben sollte. Diese Überlegungen führten zu weiteren Betrachtungen hinsichtlich der Konstruktion und Funktionalität des Systems. Es wurde dabei diskutiert, wie die Krallen effizient in das Gesamtsystem integriert werden könnten, um eine optimale Leistung zu gewährleisten. Darüber hinaus wurden verschiedene Möglichkeiten in Betracht gezogen, um die Belastbarkeit und Stabilität des Manipulators zu verbessern. Insgesamt erforderte die Umsetzung der Idee eine sorgfältige Planung und Abwägung verschiedener Faktoren, um ein funktionales und zuverlässiges System zu entwickeln.

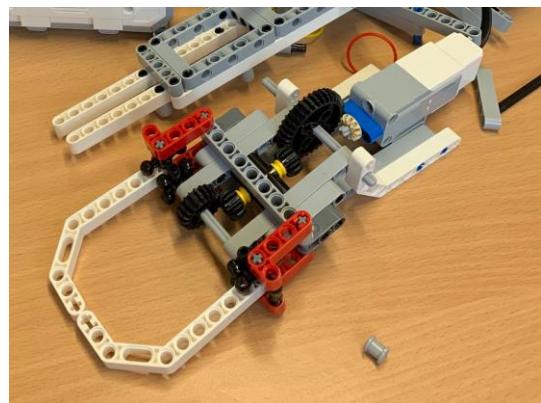


Abbildung 2: Mechanische Krallen

B. Sensortechnologie im Roboterdesign

Die Wahl der Sensoren war einer der wichtigsten Punkte, da der Ultraschallsensor (Abbildung 3) nicht verwendet werden

konnte, weil die Krallen einen Schatten auf die Sensoren warfen und er nicht korrekt arbeiten konnte, und der Touchsensor konnte eine leichte Last nicht erkennen.



Abbildung 3: Ultraschallsensor

C. Motoren im Einsatz

Die Entscheidung bezüglich der Motorwahl für die mechanischen Krallen war entscheidend für das potenzielle Gewicht der zu hebende Last. Gleichzeitig war es nicht möglich, jeden Mechanismus am Antriebsteil des Roboters anzubringen. Aus diesem Grund wurde sich für einen Motor mittlerer Größe (Abbildung 4) entschieden.



Abbildung 4: Mittelmotor

III. ENTWICKLUNGSPROZESS

A. Konstruktion

Die mechanischen Krallen (Abbildung 2) setzen sich aus einem Mittelmotor zusammen, der ein ausgeklügeltes Getriebesystem antreibt. Dieses System ermöglicht eine präzise Bewegung der Krallen um 170 Grad nach oben oder unten sowie das Öffnen und Schließen zu einem genau definierten Zeitpunkt. Durch das implementierte Getriebesystem wird die Energieeffizienz des Roboters erheblich verbessert. Im Vergleich zu einem zweimotorigen Krallen, bei der ein Motor für das Anheben und der zweite für das Öffnen und Schließen zuständig ist, entfällt hier die Notwendigkeit, den zweiten Motor anzuheben. Dies trägt dazu bei, den Energieaufwand bei der Bewältigung des zu hebenden Gewichts zu optimieren. Eine Herausforderung bestand dennoch darin, dass mechanische Krallen mit nur einem Motor eine präzise Justierung des

Getriebes erfordern, um die Öffnung im exakten Moment zu gewährleisten.

Als Lasterkennungssensor (Abbildung 5) wurde der Touchsensor gewählt, da er am stabilsten ist.



Abbildung 5: Lasterkennungssensor

Um die Vielseitigkeit des Roboter-Manipulators zu erhöhen, wurde ein Schwenkmechanismus integriert. Hierbei wurde der Roboter in zwei Komponenten aufgeteilt. Der erste Teil beherbergt den EV3-Stein, den Ultraschallsensor (Abbildung 3) sowie zwei große Motoren. Der zweite Teil umfasst die mechanischen Krallen und den speziellen Behälter. Beide Teile sind mithilfe von Zahnrädern miteinander verbunden, wodurch der zweite Teil des Roboters zu einem Anhänger des ersten Teils wird.

B. Software

Die Funktionalität des Robotermanipulators wurde mit MATLAB implementiert. Der Algorithmus (Abbildung 7) beginnt mit der Einstellung der Motorleistung und dem Starten der Motoren, um den Roboter vorwärtszubewegen. Dann tritt der Algorithmus in eine Endlosschleife ein, die bis zur Aktivierung des Touchsensors andauert. Wenn der Touchsensor aktiviert wird, stoppen die Motoren, und der mittlere Motor, der zum Anheben der Krallen um einen bestimmten Winkel verwendet wird, wird gestartet. Anschließend erfolgt eine Änderung der Motorenrichtung und ihr Start mit einer anderen Leistung, um die Krallen in die Ausgangsposition zurückzubringen. Danach werden die Bewegungsmotoren mit neuen Parametern wiederhergestellt. Wenn die Entfernung vom Ultraschallsensor zum Hindernis weniger als 50 beträgt, stoppen die großen Motoren, und der Roboter dreht sich, um das Hindernis zu vermeiden. Am Ende werden die Motorleistungen auf positive Werte eingestellt, dann wartet der Roboter 3 Sekunden, und beide Motoren stoppen. Ein Programmablaufplan zur Erklärung eines Verfolgungsalgorithmus ist in Abbildung 6 dargestellt.

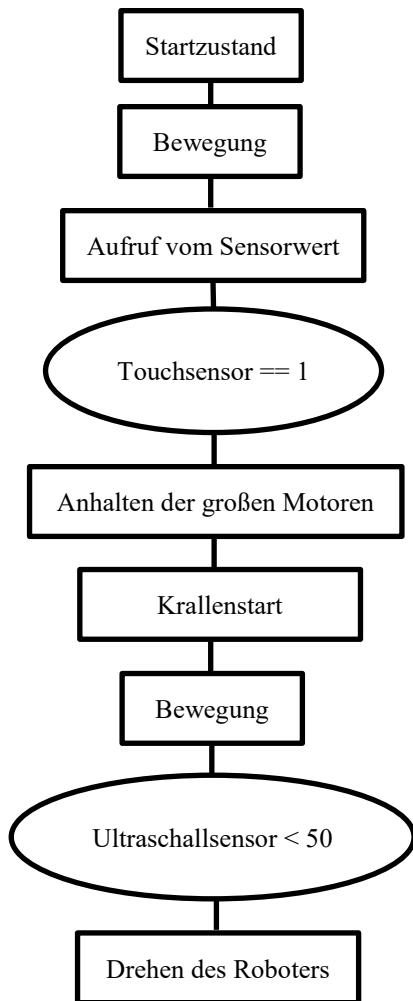


Abbildung 6: Programmablaufplan zur Erklärung eines Verfolgungsalgorithmus

IV. ERGEBNISDISKUSSION

Nach den Tests bewies der Roboter seine Leistungsfähigkeit trotz der physikalischen Einschränkungen, die mit LEGO-Steinen verbunden sind.

Das Problem, dass MATLAB nur eine Aktion auf einmal ausführen kann, wurde durch die Implementierung von `waitFor()` beseitigt. Außerdem wurden alle unnötigen Teile entfernt, was die Leistung der Maschine erheblich verbesserte.

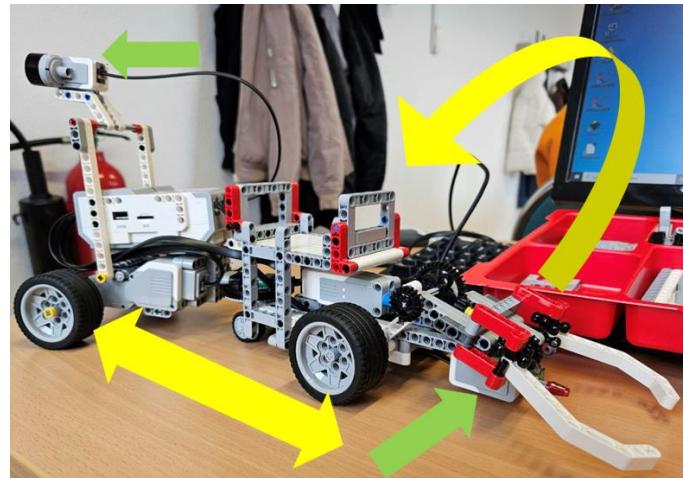


Abbildung 7: Roboter-Manipulator-Algorithmus

V. ZUSAMMENFASSUNG UND FAZIT

Zusammenfassend kann man sagen, dass alle Ziele erreicht wurden. Eine Bluetooth-Verbindung wurde ebenfalls hinzugefügt, so dass der Roboter Funktionen aus der Ferne ausführen kann. In Zukunft kann das automatische Fahrsystem verbessert und der Entladevorgang weiterentwickelt werden. Zum Beispiel könnte ein stationärer Manipulator gebaut werden, den der Robotermanipulator automatisch durch Lichtemission findet und unter ihm fährt.

LITERATURVERZEICHNIS

[1] LfB - RWTH Aachen, "Mindstorms EV3 Toolbox", Documentation, Release v1.0, Jan 27, 2020.