

Der Einparkroboter

Dennis Reiß, Elektromobilität
Otto-von-Guericke-Universität Magdeburg

Zusammenfassung—Es wurde ein autonom einparkendes Fahrzeug mit LEGO Mindstorms EV3 entwickelt. Während der Umsetzung zeigten sich mehrere Herausforderungen: Die strukturelle Toleranz der LEGO-Bausteine erschwerte die Geradeausfahrt, sodass eine Stabilisierung durch einen Gyrosensor erforderlich wäre. Zudem führte eine instabile Bluetooth-Verbindung zwischen MATLAB und dem EV3 zu Kommunikationsabbrüchen. Trotz dieser Einschränkungen erkannte das System die Parklücken und führte die Einparkvorgänge autonom aus. Verbesserungsmöglichkeiten umfassen die Integration einer Kamera zur präziseren Erkennung und eine stabilere drahtlose Kommunikation.

Schlagwörter—Autonomes Fahren, Einparkassistent, Fahrzeugsteuerung, LEGO Mindstorms, Robotik

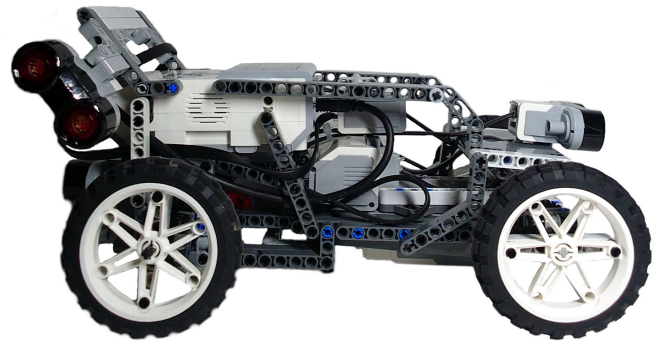


Abbildung 1. Einparkroboter

I. EINLEITUNG

SEIT der Erfindung des Benz Patent-Motorwagens vor über 200 Jahren ist das Automobil von stetigem Wandel und technischem Fortschritt geprägt [1]. Besonders in den letzten Jahren rückte die autonome Mobilität verstärkt in den Fokus von Forschung und Entwicklung.

Auch in diesem Jahr entwickelten Studierende im Projektseminar ein teilautonomes Fahrzeug, um sich mit den Grundlagen einer solchen Programmierung vertraut zu machen. Dabei standen sowohl die Mechanik als auch die Softwareentwicklung im Mittelpunkt des Projekts. Neben der Implementierung grundlegender Fahrfunktionen wurde insbesondere die sensorbasierte Umgebungswahrnehmung untersucht.

Die folgenden Abschnitte behandeln die Konstruktion des Fahrzeugs, die Programmierung sowie den Versuchsaufbau, bevor die Ergebnisse diskutiert und mögliche Verbesserungen aufgezeigt werden.

II. VORBETRACHTUNGEN

Das parallele Einparken folgt einer strukturierten Vorgehensweise, die sich in mehrere Schritte unterteilen lässt.

Zunächst wird eine geeignete Parklücke ausgewählt, die ausreichend Platz für das Fahrzeug bietet. Anschließend richtet man das eigene Fahrzeug parallel zum parkenden Fahrzeug aus, wobei etwa ein halber Meter Abstand eingehalten werden sollte. Im nächsten Schritt wird das Lenkrad vollständig nach rechts eingeschlagen, während das Fahrzeug langsam rückwärts fährt, bis das hintere Fahrzeug im Sichtfeld erscheint. Danach wird das Lenkrad geradegestellt und das Fahrzeug weiter zurückgesetzt, bis die eigene Fahrzeugfront die hintere Stoßstange des vorderen Autos passiert hat. Nachdem das Lenkrad vollständig nach links eingeschlagen wurde, um in die Parklücke einzufahren, wird das Fahrzeug zentriert und das Lenkrad in Ausgangslage gebracht [2].

III. UMSETZUNG

In diesem Projekt wurde zunächst das Fahrzeug aufgebaut, wobei grundlegende mechanische und elektronische Komponenten integriert wurden. Anschließend wurde die Steuerungsprogrammierung, sodass das Projektziel erreicht werden konnte.

A. Konstruktion

Das Design des Fahrzeugs (siehe Abb. 1) orientiert sich an klassischen Personenkraftwagen statt an typischen Robotermodellen, wodurch die Platzierung der Komponenten anspruchsvoller wurde.

Der Antrieb erfolgt über einen großen Motor, der seine Kraft über ein Differential (siehe Abb. 2) auf die Hinterräder überträgt. So werden unterschiedliche Raddrehzahlen, wie sie bei Kurvenfahrten auftreten, ausgeglichen und ein Durchdrehen der Hinterräder verhindert.

Unterhalb der Vorderachse ist ein mittlerer Motor zur elektrischen Servolenkung montiert (siehe Abb. 4). Die Kraftübertragung erfolgt über zwei Zahnräder, die eine größere Auflagefläche auf der Zahnstange der Lenkung gewährleisten. Aufgrund der Position des Servomotors war es erforderlich, größere Räder zu verbauen, um eine ausreichende Bodenfreiheit sicherzustellen.

In der hinteren Hälfte des Fahrzeugs ist der EV3-Controller fest verbaut. Dieser verfügt über jeweils vier Anschlüsse für Sensoren und Motoren. Die Sensoranschlüsse sind mit einem Gyrosensor sowie drei Ultraschallsensoren belegt, die vorne, hinten und rechts montiert sind. Während die Ultraschallsensoren zur Messung der Abstände des Autos zur Umgebung dienen, ist der Gyrosensor zwar angeschlossen, wurde aber noch nicht in die Funktionalität integriert.

Für die Verbindung zwischen dem EV3 und MATLAB ist zunächst ein Kabel erforderlich. Damit es die Räder nicht

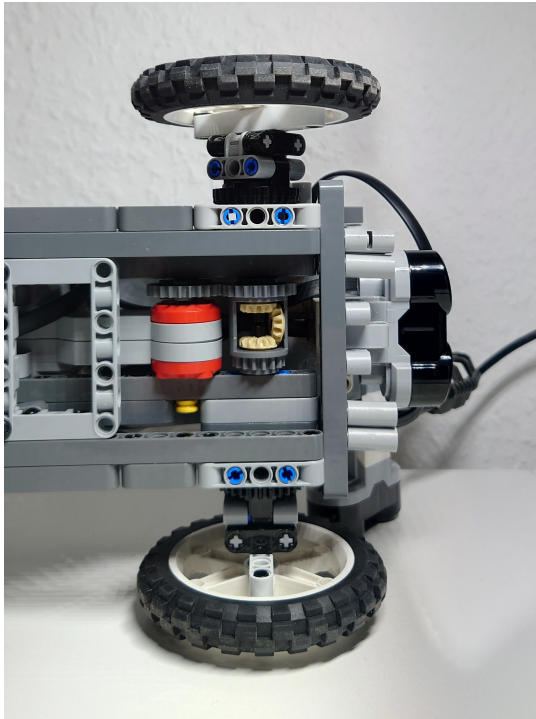


Abbildung 2. Differential der Hinterachse

blockiert, wurde es in einem Käfig oberhalb des Spoilers befestigt.

B. Programmierung

Für das Vor- und Rückwärtsfahren wurden jeweils Funktionen implementiert, die mit einer einzugebenden Geschwindigkeit und Dauer arbeiten. Um ein ungewolltes Ausrollen zu vermeiden, wurde der „brakeMode“ auf „brake“ gesetzt, sodass der Motor am Ende der Fahrt seine Position hält. Diese Einstellung wurde auch bei der Lenkungssteuerung berücksichtigt, um einen eingestellten Lenkwinkel beizubehalten.

Im Gegensatz zu den Antriebsfunktionen arbeiten die beiden Funktionen zur Lenkung nach links und rechts mit einer einstellbaren Motorleistung sowie einer Winkелеingabe. Dies ermöglicht eine Anpassung an verschiedene Fahrbahnbedingungen. Allerdings wirkt sich die Motorbremse sowohl auf die Akkulaufzeit als auch auf die Geräuschentwicklung aus, was als Nachteil in Kauf genommen werden musste.

Für die Ultraschallsensoren an der Vorder- und Rückseite wurde eine Funktion geschrieben, die bei Annäherung an ein Hindernis mit zunehmendem Abstand ein schneller werdendes akustisches Signal ausgibt und den Motor stoppt, bevor es zu einer Kollision kommen könnte.

Erfasst der Ultraschallsensor rechtsseitig einen Wert über 25 cm, wird ein boolescher Wert gesetzt, der das Vorhandensein einer Parklücke speichert. Sobald der Sensor wieder einen Wert unter 25 cm misst, wird das Ende der Parklücke erkannt und das Fahrzeug veranlasst, anzuhalten.

Auf Basis dieser Funktionen wurde die Programmierung der Parksequenz abgeleitet, wie sie in der Vorbetrachtung und in Abbildung 3 beschrieben wird.

C. Versuchsaufbau

Für die Durchführung von Tests war eine geeignete Testumgebung erforderlich. Diese bestand aus einem Teppich mit gummierter Oberfläche sowie zwei identischen Kisten. Da der Boden des Labors nicht genügend Reibung für die Fahrzeugreifen bot, wurde der Teppich als Fahrfläche gewählt. Zur Sicherstellung konsistenter Testergebnisse wurden die Positionen der Kisten markiert und für jede Versuchsdurchführung beibehalten. Die simulierte Parklücke hatte eine Länge von etwa dem Zweifachen der Fahrzeuglänge.

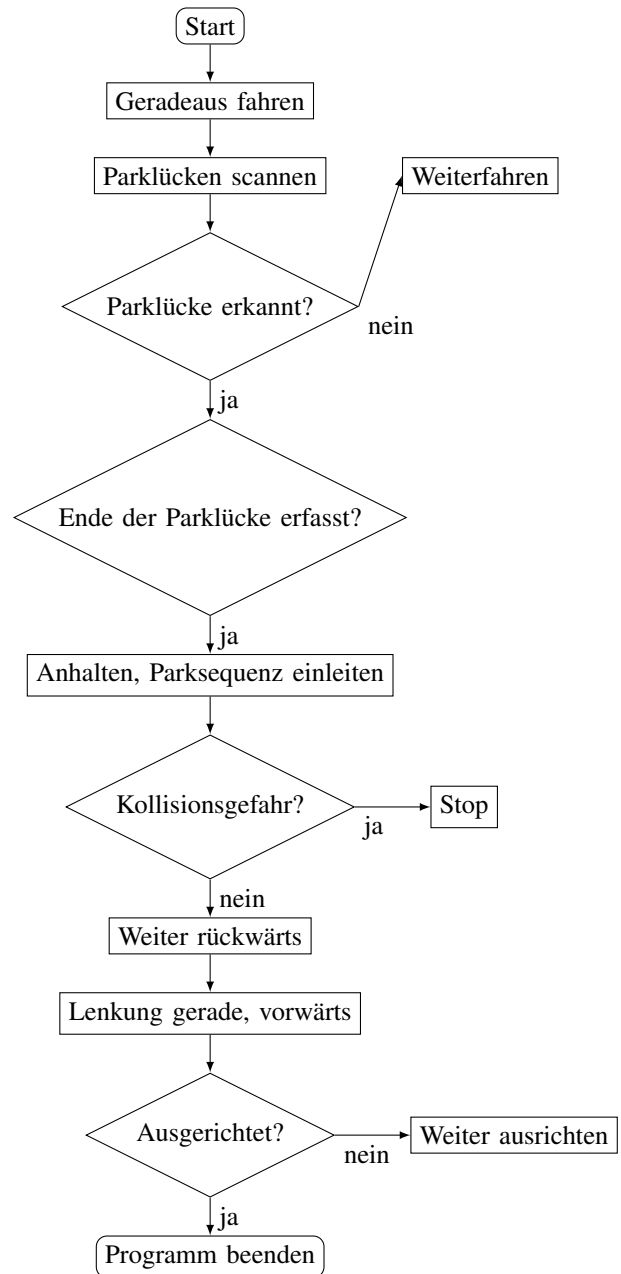


Abbildung 3. Programmablaufplan für das parallele Einparken.

IV. ERGEBNISDISKUSSION / VERBESSERUNGSVORSCHLÄGE

In der Praxis zeigte sich, dass das Fahrzeug auf längeren Distanzen keine exakte Geradeausfahrt durchführen konnte,

sondern eine leichte Kurvenfahrt vollzog. Trotz der Motorbremse in der Lenkung war eine präzise Geradeausbewegung nicht gewährleistet. Dies ist nicht nur auf die anfängliche Positionierung des Fahrzeugs zurückzuführen, sondern vor allem auf die strukturelle Toleranz der LEGO-Bausteine. Mehrere Anpassungen führten nicht zum gewünschten Ergebnis.

Ein Lösungsansatz bestünde in der Nutzung des im Fahrzeug integrierten Gyrosensors. Dieser könnte zu Beginn der Fahrt einen Referenzwert erfassen und bei Abweichungen entsprechende Lenkbefehle einleiten, um die Kursstabilität zu verbessern. Ein erster Prototyp des Programms wurde bereits entwickelt.

Geplant war die Einbindung des Gyrosensors in zwei Funktionen: Erstens zur Erkennung einer Parklücke während der Geradeausfahrt und zweitens zur präziseren Steuerung während der Einparksequenz mithilfe von Winkeln. Schon bei der ersten Funktion traten jedoch Probleme auf. Das zyklische Abfragen der Sensorwerte dauerte zu lange, sodass das Fahrzeug meist erst eine Fahrzeuglänge hinter dem Ende der Parklücke zum Stillstand kam. Dieses Problem hätte sich theoretisch lösen lassen, indem das Fahrzeug eine bestimmte Strecke zurückgesetzt hätte. In der Realität fährt man jedoch nicht so weit an einer Parklücke vorbei. Abgesehen von der Programmierung fiel zudem auf, dass der Sensorwert in einigen Fällen kontinuierlich anstieg oder sank, obwohl keine Bewegung stattfand. Eine erste Vermutung war, dass der Sensor beim Einschalten des EV3-Controllers in absoluter Ruhe sein musste. Die Einhaltung dieser Bedingung erwies sich jedoch als nicht vollständig zuverlässig. Aufgrund dieser Problematik und des zusätzlichen Zeitmangels wurde die Idee vorerst verworfen.

Eine alternative Lösung stellt der Einbau eines vierten Ultraschallsensors auf der rechten Fahrzeugseite dar. Durch die Messung der seitlichen Abstände mit zwei Sensoren könnten Korrekturen vorgenommen und eine stabilere Geradeausfahrt erreicht werden.

Während der Präsentation zeigte der Einparkroboter neben der unpräzisen Geradeausfahrt auch eine gewisse Unzuverlässigkeit in der Programmsteuerung. Eine mögliche Ursache hierfür sind die Einschränkungen der verbauten LEGO-Ultraschallsensoren. Diese liefern bei einem Abstand von weniger als 5 cm einen Wert von 255 zurück, was fälschlicherweise signalisiert, dass sich kein Hindernis in der Nähe befindet. Dies führt in der aktuellen Implementierung dazu, dass der Roboter keine Parklücke mehr erkennt und stattdessen auf das Ende der vermeintlichen Lücke wartet. Eine mögliche Lösung wäre die Integration einer Kamera, um die Erkennung der Parklücke sowie umliegender Hindernisse zu verbessern. In Verbindung mit Bildverarbeitungsalgorithmen könnten nicht nur Objekte präziser identifiziert, sondern auch Bodenmarkierungen erkannt werden, wie sie auf vielen Parkplätzen vorkommen. Dadurch ließe sich das System zuverlässiger gestalten und besser an verschiedene Umgebungen anpassen.

Ein letzter praxisnaher Verbesserungsvorschlag wäre eine Bluetooth-Verbindung zwischen MATLAB und dem EV3. Zwar lässt sich eine Verbindung zwischen Laptop und EV3 herstellen, jedoch bricht sie nach etwa zehn Sekunden ständig ab. Dies könnte auf Unterschiede in den Bluetooth-Generationen der beiden Geräte zurückzuführen sein.

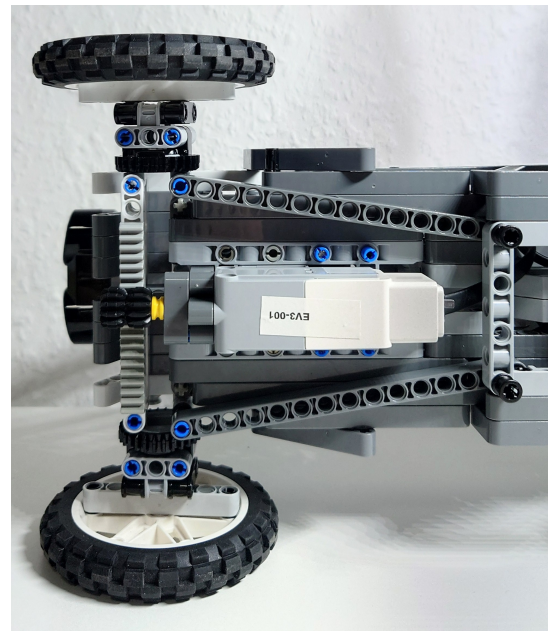


Abbildung 4. Aufbau der Lenkung

V. ZUSAMMENFASSUNG UND FAZIT

Das Ziel, einen funktionierenden Einparkroboter zu entwickeln, wurde erfolgreich erreicht. Für den Aufbau kamen zwei Motoren, drei Ultraschallsensoren und ein EV3-Controller zum Einsatz. Das zugrunde liegende Programm erkennt eine Parklücke und führt den Einparkvorgang durch, während es gleichzeitig Kollisionen vermeidet. Das Projekt hat gezeigt, dass ein autonomes Einparksystem auch mit einfachen Mitteln wie dem LEGO-Mindstorms Set realisierbar ist.

Dennoch gibt es mehrere Ansätze zur Verbesserung, auf die sich zukünftige Arbeiten konzentrieren könnten, um das System robuster und flexibler für verschiedene Umgebungen zu gestalten. Sollte das Projekt weiterentwickelt werden, sind insbesondere folgende Optimierungen empfehlenswert:

- 1) Erkennung von Parklücken sowohl auf der linken als auch auf der rechten Seite
- 2) Einsatz einer Kamera und Bildverarbeitung
- 3) Implementierung eines Spurhalteassistenten
- 4) Erhöhung der Zuverlässigkeit der Programmierung

Die bestehende Konstruktion bietet eine solide Basis für diese Weiterentwicklungen. Für Studierende höherer Semester könnte zudem die Implementierung komplexerer Algorithmen aus der Regelungstechnik eine sinnvolle Erweiterung darstellen, um das Einparksystem noch zuverlässiger zu gestalten – insbesondere unter erschwerten Bedingungen.

LITERATURVERZEICHNIS

- [1] WELT DER WUNDER: *Die Entwicklung des Autos – von den Anfängen bis heute.* <https://www.weltderwunder.de/die-entwicklung-des-autos-von-den-anfaengen-bis-heute/>. – Zugriff am: 24. Feb. 2025
- [2] ZED DRIVING SCHOOL: *How To Parallel Park In 1 Minute.* <https://www.youtube.com/watch?v=FjEmxM2PMv8>. – Zugriff am: 24. Feb. 2025