

# LABrat – Ein Roboter, der den Weg kennt

Max Bachran, Elektrotechnik und Informationstechnik  
Otto-von-Guericke-Universität Magdeburg

**Zusammenfassung**—Das diesjährige LEGO-Praktikum 2024 widmete sich der Aufgabe aus LEGO und Matlab-Programmierung einen funktionsfähigen Roboter zu bauen, der gezielte Problemstellungen überwinden kann. Dabei wurde in diesem Paper der autonom fahrende Labyrinthroboter und seine Entwicklung betrachtet. Mit Omni-Wheels, die das Fahren in alle Richtungen ermöglichen und Ultraschallsensoren zur Distanzmessung wurde ein intelligentes Fahrzeug gebaut, das sich durch ein Labyrinth bewegt und den Ausgang findet.

**Schlagwörter**—Self-driving, Hinderniserkennung, Labyrinth, Omni-Wheels, Wegfindung

## I. EINLEITUNG

Die heutige Welt wird zunehmend stärker vernetzt und digitalisiert. Dies ist in vielen Bereichen des Lebens deutlich spürbar, auch in der Automobilbranche. Dort wird seit langem an selbstfahrenden Fahrzeugen geforscht und erste Schritte, wie zum Beispiel Parkassistent oder automatische Notbremssysteme, sind schon längst für den Verbraucher erhältlich. Der Mensch wird in diesen Bereichen zunehmend von der Technik unterstützt und entlastet. Auch die vollständige Automatisierung des Fahrzeugs scheint nicht allzu weit in der Zukunft zu liegen. Die Entwicklung dieser Systeme ist im Fokus vieler und es wird aus vielen Richtungen geforscht. Aufgrund der potenziell verheerenden Konsequenzen von Fehlern für Verkehrsteilnehmer sind die sicherheitstechnischen Bedenken für viele Menschen in Bezug auf selbstfahrende Autos noch zu groß und Vertrauen zu gering. Sollten jedoch autonom fahrende Fahrzeuge zuverlässiger sein als menschliche Fahrer, könnten sie in Zukunft auf den Markt kommen und den Straßenverkehr sicherer und effizienter gestalten.

Auf Basis dieser Überlegungen entstand die Idee, bereits heute ein Fahrzeug zu entwickeln, das selbstständig fahren kann und auf äußere Umstände reagiert, damit schon heute ein Blick ins Morgen geworfen werden kann.

## II. VORBETRACHTUNGEN

### A. Welche Funktionen muss das Fahrzeug beherrschen?

Das Fahrzeug muss grundlegend in der Lage sein, sich fortzubewegen. Es sollte daher zu Vorwärts-, Rückwärts- und Seitwärtsbewegung in der Lage sein. Da das Labyrinth mit rechten Winkeln konstruiert wurde, ist diese Fahrimplementierung theoretisch ausreichend. Nur für kleineres Ausrichten muss sich das Fahrzeug drehen können.

Darüber hinaus muss es in der Lage sein, auf Hindernisse zu reagieren und daher die Ultraschallsensoren ansprechen zu können. Nun muss auf die Rückgabe der Sensoren auch reagiert werden. Eine Logik muss implementiert werden, die

auf die Hindernisse reagiert und es dem Fahrzeug ermöglicht, den richtigen Weg zu finden.

### B. Welche Komponenten werden dafür benötigt?

Für die Umsetzung des Projekts wurde ein LEGO-Bausatz bereitgestellt, der alle benötigten Komponenten für die Konstruktion enthielt. Dabei spielten einige Bauelemente eine zentrale Rolle. Der Kern der Konstruktion war der Steuercomputer, der für die Ansteuerung der weiteren Komponenten zuständig war. Zu Beginn wurde ein NXT-Baustein verwendet, der später durch einen EV3-Baustein ersetzt wurde. An diesen Baustein waren insgesamt vier Motoren und vier Ultraschallsensoren angeschlossen.

Die Ultraschallsensoren dienten der Hinderniserkennung, indem sie kontinuierlich Ultraschallsignale aussenden und deren Reflexionen an der Oberfläche von Hindernissen die Distanz zum nächsten gerade vor ihnen liegenden Hindernis ermittelten. Vier Motoren wurden als Antrieb für die vier verwendeten Räder eingesetzt. Dadurch war es möglich, alle vier Räder gleichzeitig oder separat anzusteuern. Bei den Rädern handelte es sich um spezielle Omni-Wheels [1], bei denen die Lauffläche teilweise mit Rollen ausgestattet war, deren Drehachsen im rechten Winkel zur Drehachse des jeweiligen Hauptrades lagen. Diese ermöglichten dem Fahrzeug das seitwärts Fahren, ohne dass eine Lenkbewegung erforderlich war. Alle Teile wurden mithilfe von LEGO-Steinen miteinander verbunden, um so eine stabile Konstruktion als Ganzes zu gewährleisten.

## III. UMSETZUNG

### A. Konstruktion 1

Die anfängliche Idee, das Fahrzeug in die Höhe zu bauen, um die Breite zu minimieren und somit das Labyrinth nicht zu vergrößern, erwies sich als Herausforderung. Der NXT-Baustein als Steuercomputer wurde zentral platziert, mit je einem Motor an der Längs- und Querseite, wobei die Drehachsen unterhalb des Bausteins positioniert waren. Jeder Motor sollte zwei Omni-Wheels über eine verlängerte Achse antreiben.

Allerdings stieß diese Konfiguration schnell auf Probleme. Durch die Verwendung von nur zwei Motoren kam es zu einer ungleichmäßigen Lastverteilung, was die Stabilität des Fahrzeugs beeinträchtigte. Zudem führte die Anordnung der Räder auf gleicher Höhe dazu, dass die Radachsen miteinander kollidierten. Angesichts dieser Schwierigkeiten wurde dieser Ansatz rasch verworfen, und es war klar, dass eine neue Lösung gefunden werden musste, um das Fahrzeug stabiler und funktionaler zu gestalten.

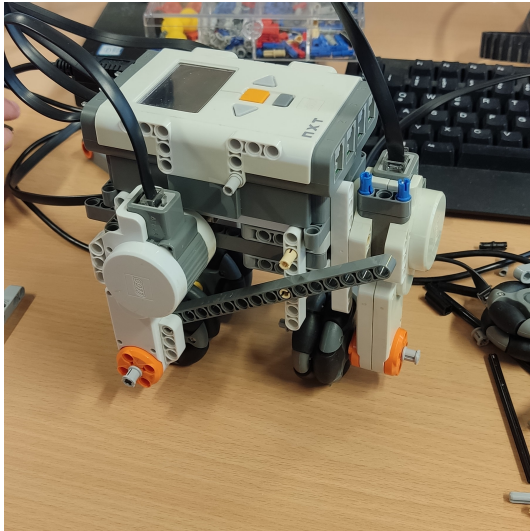


Abbildung 1. Konstruktion 2 – LABrat mit drei Motoren &amp; 3 Rädern

### B. Konstruktion 2

Aufgrund der Erkenntnisse aus der ersten Konstruktion wurden bei der zweiten Verbesserungen vorgenommen. Die Motoren wurden weiterhin senkrecht nach unten aufgestellt, wobei am NXT-Baustein zwei Motoren an der Längsseite und ein Motor am Heck an der Querseite angebracht wurden. Wie in Abbildung 1 zu sehen ist, betrieb jeder Motor ein Omni-Wheel. Diese Anordnung führte zu einer verbesserten Lastverteilung durch den dritten Motor und das dritte Rad.

Trotz dieser Verbesserungen traten beim Vorwärts- und Rückwärtsfahren horizontale Kräfte auf, die durch die Trägheit der seitlichen Motoren verursacht wurden. Diese verschoben sich während der Fahrt in die entgegengesetzte Richtung. Besonders beim seitlichen Fahren wurde dieses Phänomen beim hinteren Motor deutlich. Während es möglich war, die Motoren an der Längsseite zu stabilisieren, gestaltete sich dies beim hinteren Motor problematisch. Die auftretenden Kräfte konnten nicht vollständig unterbunden werden, obwohl rudimentäre Fahrttests mit Vorwärts- und Rückwärtsbewegungen erfolgreich durchgeführt wurden.

Weiterhin war die fehlende gemeinsame Drehachse zwischen den drei Motoren-Rad-Kombinationen ein Problem. Aufgrund der Art der LEGO-Steine war es äußerst schwierig, alle Motoren symmetrisch zueinander auszurichten. Dies führte zu unerwünschten Lenkbewegungen beim seitlichen Fahren und machte eine weitere Entwicklung dieser Konfiguration unpraktikabel.

### C. Konstruktion 3

Die dritte Konstruktion war auch das endgültige Fahrzeug. Hierbei wurde nun der NXT-Baustein durch einen EV3-Baustein ersetzt. Bei diesem konnten vorteilhafter Weise jetzt vier Motoren und dadurch vier Räder angesteuert werden. An einer Stützkonstruktion wurden sowohl der EV3-Steuercomputer als auch die Motoren befestigt, sodass sich die Drehachsen unterhalb des EV3-Bausteins befanden. Jeder Motor wurde mit einem passenden Omni-Wheel ausgestattet. Das Stützkonstrukt

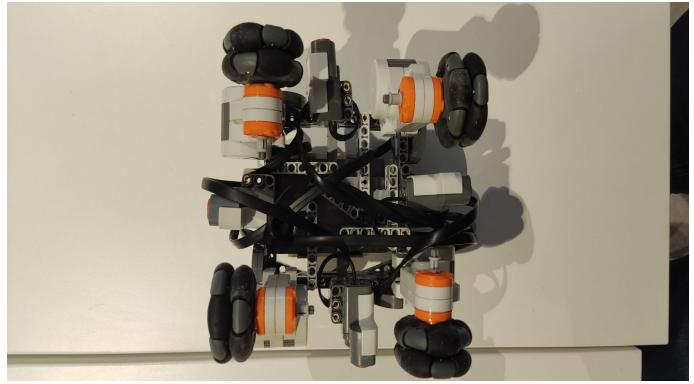


Abbildung 2. Unterseite des Fahrzeuges mit Kennzeichnung der Drehachsen

fungierte als eine Art Chassis und stabilisierte das Gesamt-konstrukt in gewissem Maße, während es auch verschiedene Anknüpfungspunkte für weitere Komponenten bot.

Die Motoren waren versetzt zueinander orientiert, wobei jeweils zwei Räder parallel zur Längsseite und zwei Räder parallel zur Querseite des EV3-Steuercomputers ausgerichtet waren, wie in Abbildung 2 zusehen ist. Durch die Symmetrie der Motoren zueinander entstand eine gemeinsame Drehachse, was ungewollte Lenkbewegungen verhinderte. Das Fahren in allen relevanten Richtungen wurde dadurch deutlich verbessert im Vergleich zu den vorherigen Konstruktionen.

Die Verwendung des EV3-Bausteins erforderte eine Anpassung des bereits geschriebenen Codes, da die Programmierung hier objektorientiert erfolgte. Der Code konnte aber problemlos migriert werden und erste Fahrttests erfolgten ohne Probleme. Anschließend wurden die vier Ultraschallsensoren angebracht, je zwei an der Längs- und Querseite des Fahrzeugs in der Mitte. Auch hier verliefen die ersten Funktionstests erfolgreich.

Mit der Fertigstellung des in Abbildung 3 zusehenden Prototypen verlagerte sich der Fokus auf das Verfeinern des MATLAB-Codes. Während der Entwicklung der finalen Konstruktion wurden kontinuierlich Funktionstests durchgeführt und der Code aus den vorherigen Revisionen an das EV3-konforme Format angepasst.

### D. Funktionsweise

In der Abbildung 4 wird der Programmablauf des Fahrzeugs vereinfacht dargestellt. Das Fahrzeug befindet sich zum Start bereits im Labyrinth und muss nun herausfinden.

Dabei hält sich das Fahrzeug stets an der rechten Wand, um garantiert den Ausgang zu finden. Da das Fahrzeug mit vier Ultraschallsensoren zur Distanzmessung ausgestattet ist, kann es die Wände in alle Richtungen kontinuierlich erfassen.

Zuerst wird das Programm initialisiert. Dabei muss beachtet werden, dass verschiedene Werte korrekt eingestellt werden, damit z.B. das Fahrzeug rechtzeitig auf Hindernisse reagiert. Zu groß und das Fahrzeug bleibt zu weit von Wänden entfernt und kann möglicherweise einen potenziellen Gang nicht erkennen. Zu klein und das Fahrzeug bremst nicht früh genug.

Dann folgt die Hauptschleife des Programms. In dieser werden die Sensoren abgefragt. Wenn ein kritischer Schwellwert unterschritten wird, weiß das Fahrzeug, dass es kurz davor

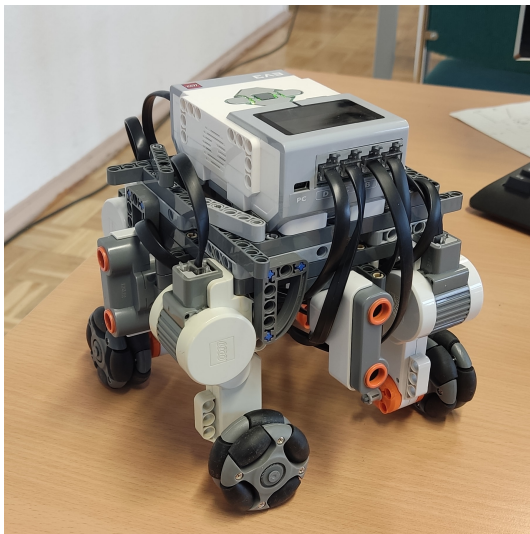


Abbildung 3. Konstruktion 3 – Finaler Prototyp der LABrat

ist in ein Hindernis zu fahren und stoppt alle Motoren. Danach überprüft es die Parallelität zur Wand durch eine Differenzberechnung zwischen den letzten Werten und den Aktuellen der seitlichen Sensorik. Bei einer Differenz muss nun das Fahrzeug die Motoren stoppen und sich ein kleines Stück drehen.

Als nächster Schritt wird die optimale Fahrtrichtung neu berechnet, um auf neu auftretende Abzweigungen reagieren zu können. Aufgrund der Prämisse sich rechts zu halten, haben effektiv nur neu auftretende Gänge rechts vom Fahrzeug einen Einfluss auf das Fahrverhalten. In diesem Fall muss das Fahrzeug vor dem Abbiegen noch etwas weiter fahren, da die Sensorik mittig vom Fahrzeug angebracht ist und bei einem sofortigen Richtungswechsel der hintere Teil des Fahrzeugs in die Wand fahren würde. Daher wird ein Moment gewartet, bevor die Motoren auch hier gestoppt werden. Es muss hier aber auch beachtet werden, dass nach diesem Richtungswechsel das Fahrzeug noch in der Kreuzung steht und soeben in Fahrtrichtung rechts der Ausgangsrichtung entspricht. Daher ist es nötig einen erneuten Richtungswechsel zu unterbinden und erst wieder zu erlauben, nachdem die Kreuzung verlassen wurde und sich das Fahrzeug im neuen Gang befindet.

Potenziell muss das Fahrzeug inzwischen die Motoren neu konfigurieren, da sich die Richtung des Fahrzeugs ändern soll oder wieder starten, weil sie gestoppt wurden. Die Motoren werden, falls sie noch laufen, gestoppt, rekonfiguriert und dann neu gestartet.

Die Motoren sind so konfiguriert, dass sie sich bis zu einem Stoppsignal drehen, um unnötige Start- und Stoppsequenzen zu vermeiden, da bei diesen sich das Fahrzeug drehen kann. Um zusätzliche Drehungen zu vermeiden, werden die Motoren aus diesem Grund auch synchron gestartet.

Da in jedem Durchlauf auch die Richtung neu berechnet wurde, fährt das Fahrzeug immer in die optimale Richtung. Solange der Ausgang nicht erreicht wurde, läuft das Programm in dieser Schleife ab. Nach Erreichen des Ausgangs kann das Fahrzeug nun ruhen.

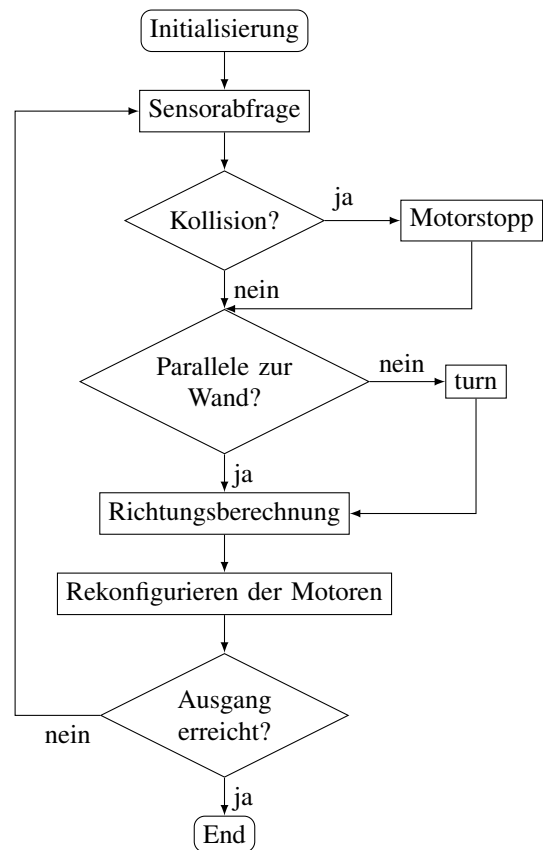


Abbildung 4. Beispielhafter Programmablaufplan zur Erklärung eines Verfolgungsalgorithmus

#### IV. ERGEBNISDISKUSSION

Es hat sich gezeigt, dass die Idee des Labyrinthroboters prinzipiell funktioniert. Bewegung und Algorithmik werden im Kern ausgeführt, wie angedacht. Allerdings gibt es noch ungeklärte Probleme in der Skalierung. Auf kleineren Teststrecken zeigt das Fahrzeug ein zuverlässiges Fahrverhalten. Aber je größer das Labyrinth ist, desto häufiger treten Fehler auf. Drei Probleme konnten dabei charakterisiert werden.

Das Erste ist die Umgebung. Der Boden und seine Beschaffenheit beeinflussen, wenn auch nur geringfügig, das Fahrverhalten des Fahrzeugs.

Zudem ist die aktuelle Konstruktion, genauer gesagt die LEGO-Bausteine, problematisch. Es konnte festgestellt werden, dass horizontale Kräfte auf das Fahrzeug wirken, wenn dieses sich vorwärts, rückwärts oder seitwärts bewegt. Diese Kraft hat Auswirkungen auf die Motoren und ungewollte Bewegungen, welche die Richtung des Fahrzeugs beeinflussten. Ein Drift nach links oder rechts der Fahrtrichtungen war zu beobachten. Auch wenn versucht wurde diesen auszugleichen, hat dies nur mittelmäßig funktioniert.

Des Weiteren sind die Ultraschallsensoren nur in der Lage gerade vor ihnen zu messen. Das Fahrzeug kann nur erkennen, ob Hindernisse direkt vor den Sensoren sind. Daher entstehen sehr große tote Winkel, in denen das Fahrzeug blind ist. Daher kommt es in Extremfällen zu Kollisionen. Durch präzise Parameterbestimmung kann dieses Problem minimiert werden, allerdings sind diese nur für eine Teststrecke optimiert und

führen auf neuen Strecken erneut zu Problemen.

#### V. ZUSAMMENFASSUNG UND FAZIT

Zusammenfassend lässt sich sagen, dass kreative Projekte mit LEGO und MATLAB verwirklicht werden können. Die Implementation von verschiedenen Funktionen zum Fahren, zur Hinderniserkennung und Algorithmik lässt sich schnell lernen und umsetzen. Selbst ein selbständig fahrendes Fahrzeug kann entwickelt werden.

Festzuhalten ist, dass jedes Projekt stets mit einem realisierbaren Konzept angegangen werden sollte, im kleinen angefangen werden muss und dann nach erfolgreichen Tests skaliert werden kann.

In Bezug auf den Labyrinthroboter kann gesagt werden, dass die Konstruktion noch Schwachstellen beinhaltet, die eine höhere Stabilität und Berechenbarkeit zurückhalten. Auch ist der Code sehr Parameter und Labyrinth abhängig, weshalb entweder mehr Sensoren oder eine andere Herangehensweise in Betracht gezogen werden sollten.

#### LITERATURVERZEICHNIS

- [1] <https://de.wikipedia.org/wiki/Allseitenrad>