

„Mindcub4r“ mit Lego Mindstorms

Anton Kresan, Elektrotechnik und Informationstechnik
Otto-von-Guericke-Universität Magdeburg

Zusammenfassung—Jedes Jahr findet bei Otto-von-Guericke Magdeburg ein Design - Workshop statt, bei dem Studierende der Ingenieurwissenschaften einen Roboter aus Lego entwerfen.

Im Rahmen des diesjährigen Projekts wurde ein spezieller Mechanismus entwickelt – der „Mindcub4r“. Dabei handelt es sich um einen Roboter, der gezielte Manipulationen am Rubik's Cube durchführt.

Dieses Projekt zielt darauf ab, zu untersuchen, wie der Roboter mit komplexeren Manipulationen umgehen kann. Durch präzise programmierte Bewegungsabläufe, insbesondere den Pif-Paf-Algorithmus, wird getestet, wie stabil und genau der Roboter die geplanten Sequenzen ausführt. Für die Gestaltung kamen Lego -Mindstorms-Sets und der EV3 -Steuerungscomputer zum Einsatz. Dieser Artikel beschreibt die Struktur und Funktionen des Mechanismus. Auch die bei der Gestaltung aufgetretenen Probleme und deren Lösungen wurden erwähnt.

Schlagwörter—Rubik's Cube, LEGO Mindstorms, MATLAB, Robotik, Pif-Paf-Algorithmus.

I. EINLEITUNG

DIE Manipulation von Objekten durch autonome Roboter ist ein wichtiger Bereich der modernen Robotik. Der Rubik's Cube stellt dabei eine ideale Testplattform dar, da er sowohl präzise motorische Bewegungen als auch eine exakte Steuerung erfordert.

Das Ziel dieses Projekts war die Entwicklung eines Roboters, der kontrollierte Manipulationen am Rubik's Cube ausführen kann. Hierbei wurde der Pif-Paf-Algorithmus (Abbildung 1), als zentrale Bewegungssequenz implementiert. Diese Bewegung ist eine der fundamentalen Sequenzen zur Lösung des Würfels und dient als Basis für viele fortgeschrittene Algorithmen. Durch den Einsatz von LEGO Mindstorms EV3 und MATLAB wurde eine effiziente Steuerung des Roboters realisiert, die eine präzise und wiederholbare Ausführung der Bewegungen gewährleistet.

II. VORBETRACHTUNGEN

In diesem Abschnitt werden die zentralen Werkzeuge dieses Projekts vorgestellt und ihre Funktionen im Zusammenhang mit der Durchführung spezifischer Aufgaben oder der Reaktion auf Eingaben beschrieben.

A. Programmierungsumgebung

Die Technologie basiert auf LEGO -Bausteinen, elektronischen Komponenten und spezialisierter Software, die zur Entwicklung und Steuerung autonomer Roboter eingesetzt wird.

Im Rahmen des Projekts wurden zwei mögliche Steuerungssysteme in Betracht gezogen: LEGO Mindstorms EV3 und dessen Vorgängermodell NXT. Nach einer umfassenden

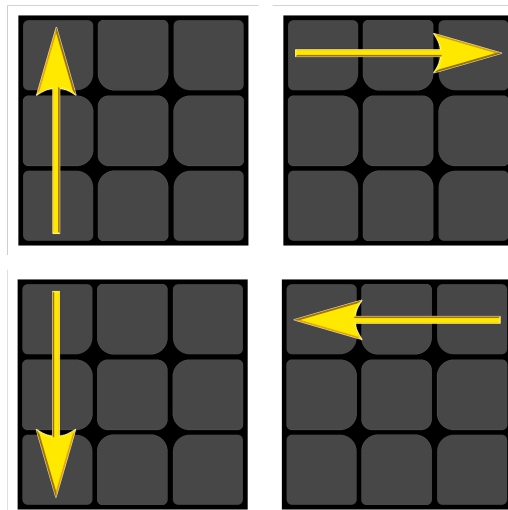


Abbildung 1. Pif-Paf-Algorithmus [1]

Analyse der technischen Spezifikationen und Anforderungen des Projekts fiel die Wahl auf EV3, da es über eine höhere Rechenleistung, präzisere Motorsteuerung und erweiterte Sensorik verfügt. Die Motorik ist präziser und weist eine höhere Auflösung bei Winkelbewegungen auf, was fließende Bewegungen und Drehungen ohne Fehler ermöglicht. Diese Eigenschaften sind entscheidend für die exakte und wiederholbare Durchführung der Manipulationen am Rubik's Cube.

B. MATLAB als Entwicklungsumgebung

Die Wahl der Programmiersprache spielt eine entscheidende Rolle für die effiziente Steuerung und Entwicklung von Robotersystemen. Für das vorliegende Projekt wurde MATLAB als Hauptprogrammiersprache gewählt, da es eine direkte und leistungsstarke Schnittstelle zur Steuerung des LEGO Mindstorms EV3 bietet.

MATLAB bietet leistungsstarke Funktionen für Vektorrechnung, Matrizenoperationen und Algorithmenoptimierung, die nicht nur bei der Umsetzung präziser Bewegungen von Vorteil sind, sondern auch für die exakte Steuerung eines Roboters und die Speicherung von Daten über den Zustand des Rubik's Cubes unerlässlich sind. Darüber hinaus ermöglicht MATLAB die separate Programmierung von Sensoren und Motoren, wodurch der Funktionsumfang des Roboters flexibler gestaltet und die Implementierung der Steuerungsalgorithmen erheblich vereinfacht wird.

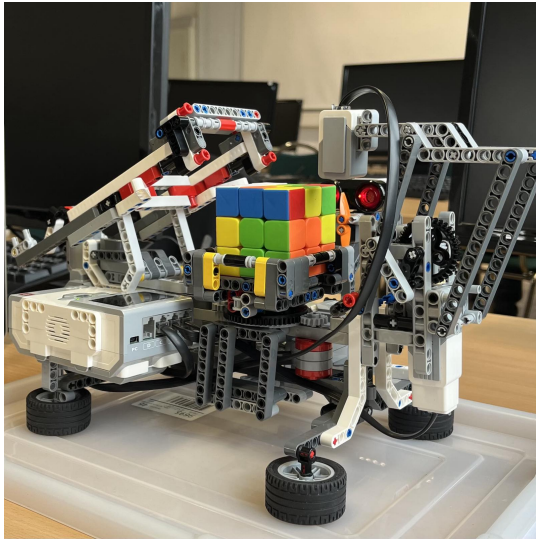


Abbildung 2. Mindcub4r [2]

III. KONSTUKTION UND PROGRAMMIERUNG

A. Aufbau

Die Konstruktion des Roboters Mindcub4r basiert auf einer durchdachten und zuverlässigen mechanischen Struktur, die eine präzise Steuerung aller Komponenten gewährleistet.

Die Anordnung der Motoren und Sensoren wurde so gestaltet, dass gegenseitige Beeinträchtigungen vermieden werden, um einen effizienten und reibungslosen Bewegungsablauf zu gewährleisten. Zudem wurden vier zentrale Mechanismen entwickelt, um die vollständige Funktionalität des Roboters sicherzustellen und eine präzise Manipulation des Rubik's Cubes zu ermöglichen. (Abbildung 2).

Im Zentrum befindet sich eine Plattform, auf der Rubik's Cube fixiert und gedreht wird. Links wurde ein Greifmechanismus integriert, der die anspruchsvollsten Manipulationen ausführt, darunter das Umklappen und Fixieren des Würfels. Rechts befindet sich ein ausfahrbarer Mechanismus mit einem Farbsensor, der zur Farbenerkennung genutzt wird. Ergänzend wurde auf der Rückseite ein Ultraschallsensor platziert, der die Positionserkennung unterstützt. Diese kompakte und stabile Konstruktion ermöglicht eine präzise Steuerung und effiziente Manipulation des Würfels.

1) *Motoren:* Wie bereits zuvor erwähnt, spielen die Motoren eine zentrale Rolle in der Funktionsweise des Roboters. Zwei große Motoren (Abbildung 3) sind für die Rotation der Plattform sowie die Bewegungen des Greifmechanismus verantwortlich. Ein mittlerer Motor (Abbildung 4) steuert den ausfahrbaren Mechanismus mit dem Farbsensor. Diese gezielte Ansteuerung der Motoren gewährleistet eine präzise und zuverlässige Durchführung der Manipulationsprozesse.

2) *Ultraschallsensor Sensor:* Die gesamte Programmsequenz startet erst, wenn der Ultraschallsensor (Abbildung 5) bestätigt, dass der Rubik's Cube korrekt auf der Plattform liegt. Er fungiert als zentrales Kontrollsystem, das die Startbedingung überwacht und erst nach erfolgreicher Erkennung die Manipulationsabläufe freigibt. Dadurch wird sichergestellt, dass alle Bewegungen präzise und fehlerfrei ausgeführt werden.

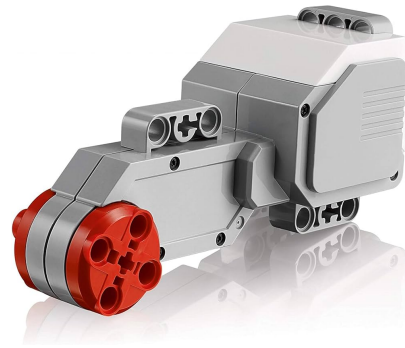


Abbildung 3. Großer Motor [3]



Abbildung 4. Mittlerer Motor [4]

3) *Farbsensor:* Der Farbsensor (Abbildung 6) fungiert als Informationsüberträger, der sämtliche Daten über die einzelnen Flächen des Rubik-Würfels an den Mikrocontroller weiterleitet. Basierend auf diesen erfassten Informationen und unter Anwendung des im Code implementierten Algorithmus führt der Roboter präzise Manipulationen am Würfel durch. Diese Datenverarbeitung ermöglicht eine gezielte Steuerung der Bewegungsabläufe und gewährleistet eine exakte Umsetzung der programmierten Sequenz.

B. Programmierung

Die korrekte und präzise Funktionsweise des Roboters hängt nicht nur von seiner mechanischen Konstruktion, sondern auch von einer strukturierten und fehlerfreien Programmierung ab. Um eine klare und systematische Steuerung zu gewährleisten,



Abbildung 5. Ultraschallsensor [5]



Abbildung 6. Farbsensor [6]

wurde der Code für jede Bewegung jedes Mechanismus separat programmiert. Besondere Steuerungsalgorithmen wurden für die Sensoren entwickelt, um eine präzise Datenerfassung und Verarbeitung zu ermöglichen. Ein herausragendes Beispiel ist der Farbsensor, dessen erfasste Daten in sechs 6×6-Matrizen überführt werden. Diese Matrizen repräsentieren die sechs Seiten des Rubik's Cubes, wobei jeder einzelne Würfelstein eine spezifische Position innerhalb der Matrix einnimmt. Die gescannten Farbinformationen werden in der MATLABKonsole visualisiert, wodurch eine genaue Analyse und Weiterverarbeitung der Daten ermöglicht wird. Unten ist der Piff-Paff-Algorithmus. Jede Zeile des Algorithmus ist eine separate Manipulation, für die ein separater Code geschrieben wird (Abbildung 7).

Aus dem Algorithmus ist ersichtlich, dass nur 2 Motoren beteiligt sind. Beispielsweise hat Motor B separate Codes vom Algorithmus. Die Programme „turnWhenHold“ (Abbildung 8) und „turnLeftWhenHold“ sind dafür zuständig, die Plattform mit dem Zauberwürfel um jeweils 90 Grad nach rechts bzw. links zu scrollen. Man kann auch feststellen, dass im Programmalgorithmus Motor B viermal vorkommt. Das bedeutet, dass die Ränder so oft nach rechts oder links gescrollt werden müssen, um einen Piff Paff zu erhalten.

Programme mit Motor C sind ausschließlich für die Bewegungen der Greifzange zuständig. Das Programm „hold“ sorgt dafür, dass die Greifzange den Würfel fixiert. Das Programm „flipAfterHold“ dreht den Zauberwürfel wiederum mit umgekehrter Geschwindigkeit in eine andere Position. Diese Programme werden mit dem Programm für die Plattform kombiniert. Dadurch wird der Zauberwürfel fixiert, die Plattform dreht die Fläche, und anschließend kippt die Greifzange den Zauberwürfel um. Das Programm „flipOver“ (Abbildung 9) führt die Befehle „hold“ und „flipAfterHold“ zusammen aus.

C. Probleme

Die Entwicklung eines Roboters zum Lösen des Rubik's Cube brachte unerwartete technische Herausforderungen mit sich. Obwohl der Roboter über alle notwendigen Mechanismen zur Manipulation des Würfels verfügte, traten erste Schwierigkeiten bei der Farberkennung auf. Bei Tests mit zwei Farbsensoren stellte sich heraus, dass vier von sechs Farben fälschlicherweise als Blau erkannt wurden. Versuche, die Farberkennung durch Anpassung der Wertebereiche zu verbessern, führten nur zu

```
clear all;
ev3 = legoev3('usb');
motorC = motor(ev3, 'C');
motorB = motor(ev3, 'B');
for i = 1:6
    hold(motorC);
    turnWhenHold(motorB);
    flipAfterHold(motorC);
    hold(motorC);
    turnWhenHold(motorB);
    flipAfterHold(motorC);
    flipOver(motorC);
    flipOver(motorC);
    hold(motorC);
    turnLeftWhenHold(motorB);
    flipAfterHold(motorC);
    hold(motorC);
    turnLeftWhenHold(motorB);
    flipAfterHold(motorC);
    flipOver(motorC);
    flipOver(motorC);
end
```

Abbildung 7. Algorithmus

```
function turnWhenHold(motorB)
    pause(0.2);
    resetRotation(motorB);
    motorB.Speed = -49.5;
    start(motorB);
    while abs(readRotation(motorB)) < 14/16 * 360
        pause(0.4);
    end
    motorB.Speed = 0;
    pause(0.3); stop(motorB);
    resetRotation(motorB);
    motorB.Speed = 40;
    start(motorB);
    while abs(readRotation(motorB)) < 14/16*77.5
        pause(0.18);
    end
    motorB.Speed = 0;
    pause(0.1);
    stop(motorB);
    pause(1);
end
```

Abbildung 8. turnWhenHold

begrenzten Erfolgen: Während das Scannen einzelner der neun Felder auf einer Würfelseite akzeptable Ergebnisse lieferte, versagte der Sensor bei rotierender Plattform in der korrekten Identifizierung aller Farben.

Um dieses Problem zu umgehen, wurde beschlossen, den Roboter so zu programmieren, dass er standardisierte algorithmische Sequenzen, bekannt als „Pif-Paf“, ausführt, ohne auf vollständige Farberkennung angewiesen zu sein.

Zusätzlich wurde der Roboter mit einem ausfahrbaren Mechanismus mit Farbsensor ausgestattet, der ihm ermöglicht,

```

function flipOver(motorC)
motorC.Speed = -30.5;
motorC.start();
pause(2);
motorC.stop();
motorC.Speed = 27;
motorC.start();
pause(1.5);
motorC.stop();
end

```

Abbildung 9. flipOver

seine Funktionalität zu erhalten und seine Fähigkeiten zu demonstrieren.

Ein weiteres Konstruktionsproblem betraf die Plattform. Diese war zunächst auf drei kleinen Zahnrädern montiert, erwies sich jedoch als mechanisch instabil und unzuverlässig. Zur Verbesserung der Funktionssicherheit wurde die ursprüngliche Anordnung durch eine Kombination aus einem großen und zwei kleineren Zahnrädern ersetzt. Nach Umsetzung dieser Modifikation zeigte das System eine deutlich gesteigerte Zuverlässigkeit, Präzision und Betriebsstabilität.

IV. ERGEBNISDISKUSSION

Zusammenfassend lässt sich feststellen, dass der Roboter erfolgreich in der Lage ist, die "Pif-Paf"-Algorithmen eigenständig auszuführen. Für das vollständige Lösen des Rubik's Cube ist jedoch eine präzise Farberkennung jeder Seite erforderlich. Obwohl die aktuellen Farbsensoren gewisse Einschränkungen aufweisen, bietet dies eine hervorragende Gelegenheit für zukünftige Optimierungen. Mit gezielten Verbesserungen im Farberkennungssystem kann der Roboter das Potenzial entfalten, den Rubik's Cube vollständig autonom zu lösen.

LITERATURVERZEICHNIS

- [1] CUBCORNER, Dzen: *Rubic's Cube 3*3*3*. <https://dzen.ru/a/YioqZz1XWGFx5YWy>. Version: 2018. – Accessed: 2025-03-13
- [2] INSTAGRAM: *Mindcub4r*. https://www.instagram.com/ovgu_projektseminar?igsh=cWJkeG9pdzdsa3Jo. Version: 2025. – Accessed: 2025-03-13
- [3] AMAZON: *LEGO MINDSTORMS Education EV3 Servomotor groß*. <https://www.amazon.de/EDUCATION-MINDSTORMS%C2%AE-Education-Gro%C3%9Fer-EV3-Servomotor/dp/B00E1QDP4W>. Version: 2025. – Accessed: 2025-03-13
- [4] AMAZON: *LEGO MINDSTORMS Education EV3 Medium Servomotor*. <https://www.amazon.de/LEGO-MINDSTORMS-Education-Servomotor-45503/dp/B00E1Q5NBU>. Version: 2025. – Accessed: 2025-03-13
- [5] AMAZON: *LEGO® MINDSTORMS® Education EV3 Ultraschallsensor*. <https://www.amazon.de/LEGO-MINDSTORMS-Education-Ultraschallsensor-45504/dp/B00E1PTRAE>. Version: 2025. – Accessed: 2025-03-13
- [6] AMAZON: *LEGO MINDSTORMS Education EV3 Farbsensor*. <https://www.amazon.de/LEGO-MINDSTORMS-Education-Farbsensor-45506/dp/B00E1QNH2W>. Version: 2025. – Accessed: 2025-03-13