

Sudoku Solver

Zhu Ding, Elektro- und Informationstechnik
Otto-von-Guericke-Universität Magdeburg

Zusammenfassung—Im Rahmen des diesjährigen LEGO Mindstorms Praktikumsprogramms wurde ein Roboter aus LEGO Baukästen gebaut, der mit NXT gesteuert und in MATLAB programmiert wurde und in der Lage war, eigenständig Sudoku-Probleme zu lösen. Der nächste Abschnitt beschreibt den Prozess der Entwicklung des Roboters.

Schlagwörter—Schlagwörter—LEGO Mindstorms, MATLAB, Sudoku, Lichtsensor, Roboter, Projektseminar

I. EINLEITUNG

IM heutigen digitalen Zeitalter verändert sich die Welt in einem noch nie dagewesenen Tempo, da die Technologie immer weiter voranschreitet und der Einsatz von LEGO—Robotern immer mehr Beachtung und Aufmerksamkeit findet. Lego-Roboter sind nicht mehr nur Spielzeug. LEGO-Roboter können auch Ziele erreichen, die auf den Ideen der Menschen basieren. In diesem Artikel geht es darum, wie Sudoku-Probleme von LEGO-Robotern gelöst werden. Sudoku ist ein klassisches Logikrätsel, das nicht nur unser logisches Denkvermögen trainiert, sondern auch Geduld und Konzentration fördert. Menschen stoßen beim Lösen von Sudoku oft auf eine Menge Probleme, aber jetzt können diese Sudoku-Probleme durch die Entwicklung präziser Algorithmen für LEGO-Roboter gelöst werden, die diese Sudoku-Probleme lösen. Ein System, das Sudoku-Probleme automatisch löst, kann die Probleme, auf die der Spieler stößt, leicht lösen, was dem Spieler nicht nur hilft, Zeit zu sparen, sondern ihm auch ermöglicht, die Lösung des Problems aus einer anderen Perspektive zu entdecken. Deshalb wird in diesem Artikel ausführlich erklärt, wie man Sudoku-Aufgaben mit LEGO-Robotern lösen kann.

II. VORBETRACHTUNGEN

Das Ziel dieses Projekts ist ein LEGO-Roboter, der Sudoku-Probleme löst. Zunächst muss man verstehen, wie man Sudoku-Probleme löst und in der Lage sein, sie in MATLAB Code darzustellen.

A. Das unverzichtbare MATLAB

Wenn man LEGO-Roboter programmieren will, muss man MATLAB beherrschen und benutzen. Das Erlernen einer Programmiersprache erfordert zunächst die Beherrschung der grundlegenden Syntax und Variablendeclarationen sowie der grundlegenden mathematischen Operationen. Darüber hinaus geht es vor allem um das Schreiben und Aufrufen von Funktionen. In diesem Projekt wird zum Beispiel der Motor mit MATLAB-Code gesteuert. Dazu gehören die Initialisierung der Verbindungen, die Angabe des Motoranschlusses und der

Parameter sowie das Senden von Steuerbefehlen zum Starten, Stoppen oder Ändern der Bewegung des Motors. Anschließend wird der geschriebene Code in die NXT-Box hochgeladen und ausgeführt. Schließlich werden die erforderlichen Parameter angepasst und optimiert.

B. Wie man Sudoku löst

Die Lösung eines Sudoku-Rätsels erfordert die Einhaltung einer Reihe grundlegender Schritte. Zunächst ist das Verständnis der Regeln des Sudoku-Spiels entscheidend, um sicherzustellen, dass jede Zahl in jeder Zeile, jeder Spalte und jedem 3×3 -Untergitter (9×9 -Netz) nur einmal vorkommt. Anschließend werden die bekannten Zahlen überprüft und logische Schlussfolgerungsmethoden wie die Methode der einzigen möglichen Kandidaten und die Ausschlussmethode angewendet, um die leeren Zellen zu füllen. Während des Füllvorgangs wird kontinuierlich ausprobiert und überprüft, um sicherzustellen, dass die eingetragenen Zahlen nicht gegen die Regeln verstößen oder zu Widersprüchen führen. Diese Schritte werden wiederholt, bis alle leeren Zellen korrekt ausgefüllt sind und das Sudoku-Rätsel gelöst ist.

5	3		7					
6			1	9	5			
	9	8						6
8			6					3
4			8	3				1
7			2					6
	6				2	8		
			4	1	9			5
			8			7	9	

Abbildung 1. Sudoku-Tabelle

III. REALISIERUNG

A. Aufbau

Das ganze Projekt besteht aus einer NXT-Box, drei Motoren, einem Lichtsensor, einigen Teilen und einigen Antriebsrädern. Die Hauptstruktur besteht aus einem Fahrzeug, das sich hin- und herbewegen kann, und auf dem Fahrzeug befindet sich ein Ausleger, der sich von einer Seite zur anderen drehen kann, und dieser Ausleger kann angehoben und abgesenkt werden.

Abbildung 2 zeigt den anfänglichen Aufbau. Da ein wichtiges Teil fehlte Abbildung 3, wurde versucht, einige Teile als Ersatz selbst herzustellen, aber das Endergebnis ist, dass die gesamte Struktur sehr instabil ist!



Abbildung 2. ursprünglicher Aufbau des Sudoku Solvers



Abbildung 3. Zahnrad

Schließlich wurde dieses wichtige Zahnrad (Abbildung 3) erhalten und es begann, der Aufbau zu modifizieren. Der größte Nachteil der ursprünglichen Konstruktion war die strukturelle Instabilität. Erstens war das Auto darunter kleiner, dann waren die NXT-Box und der Motor schwerer, was dazu führte, dass sich die Kräfte auf das Auto in der Mitte konzentrierten, was es schwierig machte, das Gleichgewicht zu halten. Zweitens wird durch das Fehlen dieses Zahnrads (Abbildung 3) die Kraft auf den Ausleger unausgewogen, was dazu führt, dass der Ausleger als Ganzes nach unten kippt, was das Gleichgewicht des gesamten Roboters beeinträchtigt.

Abbildung 4 zeigt einen verbesserten Aufbau. Es wurde damit begonnen, die Unterseite des Autos zu modifizieren. Das Auto wurde vergrößert, um einen stabilen Boden für den gesamten Roboter zu haben, und es wurden die gegebenen Teile verwendet, um das gesamte Auto stabil zu machen, damit das Gewicht der NXT-Box und der Motoren getragen werden kann. Die unausgewogenen Kräfte auf dem Ausleger wurden auch gelöst, als das Zahnrads verwendet wurde, weil das Zahnrads die Struktur stabiler macht.

B. Durchführung

Steuern Sie zunächst den hinteren Motor, damit das Auto vorwärts fährt. Wenn der Lichtsensor den Sudoku-Tabelle betritt, halten Sie an und beginnen Sie mit dem Scannen. Steuern Sie den vorderen Motor so, dass er sich von links nach rechts dreht und den Ausleger mitbewegt. Jetzt kann der Lichtsensor den gesamten Bereich in der ersten Reihe von links nach rechts abtasten. Dann lässt man den hinteren

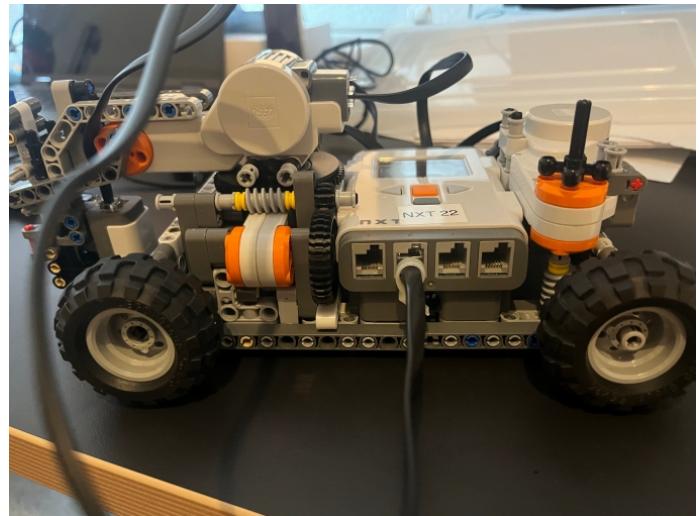


Abbildung 4. verbesserter Aufbau des Sudoku Solvers

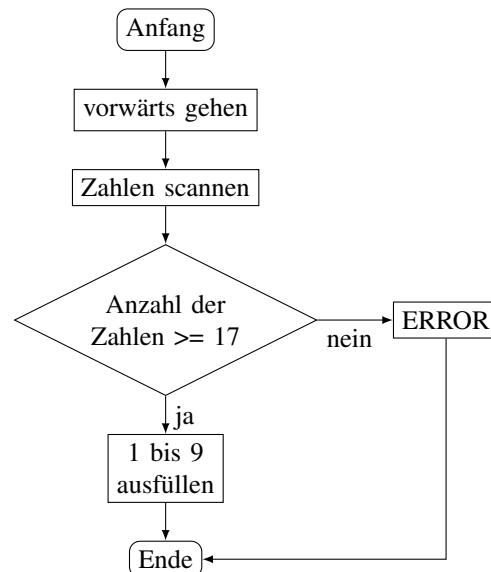


Abbildung 5. Ablaufplan

Motor das Auto ein Stück vorwärts zur zweiten Reihe fahren und wiederholt den vorherigen Vorgang, bis der Roboter alle Zahlen abgetastet hat. Normalerweise gibt es 17 Zahlen in der Sudoku-Tabelle, der Roboter sollte in der Lage sein, alle Zahlen zu scannen, um mit dem Lösen fortzufahren. Wenn die gescannten Zahlen weniger als 17 sind, wird nach dem Ende des Scanvorgangs "ERROR" auf der NXT-Box angezeigt, um uns zu informieren.

C. Probleme

Oben wurden die 17 Zahlen genannt, die vom Lichtsensor abgetastet werden müssen, um die folgenden Schritte korrekt auszuführen. Eines der Probleme, auf die gestoßen wurde, war, dass der Lichtsensor nicht alle 17 Zahlen scannen konnte. Es wurde festgestellt, dass die Ursache des Problems darin lag, dass nicht genug Licht vorhanden war. Es wurden viele verschiedene Szenarien ausprobiert, und im Grunde genommen

konnte nur eine externe Lichtquelle dazu führen, dass 3 bis 5 Zahlen gescannt wurden. Beim Versuch, eine externe Lichtquelle hinzuzufügen, konnte je nach Lichtquelle der Lichtsensor etwa 10 Zahlen oder mehr scannen. Es wurde versucht, eine Handy-Taschenlampe oder mehrere Handy-Taschenlampen oder eine Schreibtischlampe als Lichtquelle zu verwenden. Nach vielen Versuchen wurde herausgefunden, dass es am besten funktionierte, wenn nur eine Handytaschenlampe als Lichtquelle verwendet wurde, und etwa 15 Nummern gescannt werden konnten, aber immer noch nicht alle. Ein weiteres Problem ist, dass die von uns ausgedruckte Sudoku-Tabelle den Lichtsensor in die Irre führt. Die vertikalen Linien zwischen den Quadranten werden vom Lichtsensor fälschlicherweise als Zahlen 1 interpretiert, was dazu führt, dass die endgültige Anzeige der Tabelle viele Zahlen 1 enthält, was eindeutig gegen die Sudoku-Regeln verstößt und dazu führt, dass der Roboter auch die folgenden Lösungsoperationen nicht richtig ausführen kann.



Abbildung 6. Zahlen scannen

Wie auch auf Abbildung 6 zu sehen ist, wurde eine Lichtquelle von außen hinzugefügt. Als der Scan abgeschlossen war, wurden nur 13 Zahlen auf der NXT-Box angezeigt, was eindeutig nicht den Regeln des Sudoku entspricht.

IV. ERGEBNISDISKUSSION

Nachdem die Vorschläge und Genehmigungen der Professoren eingeholt worden waren, wurden die implementierten Funktionen des Roboters geändert. Der Roboter musste in die Lage versetzt werden, die Zahlen 1 bis 9 ohne Probleme auf ein leeres Blatt Papier zu schreiben, und es wurde auch versucht, "Danke" zu schreiben.

V. ZUSAMMENFASSUNG UND FAZIT

Zusammenfassend lässt sich sagen, dass während des gesamten Projekts einige Probleme auftraten und nach erfolglosen Versuchen der Rat der Professoren in Anspruch genommen wurde, um das neue Ziel zu erreichen. In diesem Projekt können auch einige Teile gefunden werden, die noch verbessert werden müssen. In Zukunft kann der Lichtsensor durch eine Webcam ersetzt werden, die alle Zahlen deutlicher und einfacher ablesen kann.

ANHANG

```

COM_SetDefaultNXT( handle );
Downie = NXTMotor( 'A' , 'Power' , 100);
Downie . TachoLimit = 250;
Downie . SendToNXT();
Downie . WaitFor();
Uppie = NXTMotor( 'A' , 'Power' , -100);
Uppie . TachoLimit = 250;
Uppie . SendToNXT();
Uppie . WaitFor();
Leftie = NXTMotor( 'C' , 'Power' , -100);
Leftie . TachoLimit = 80;
Leftie . SendToNXT();
Leftie . WaitFor();
Rightie = NXTMotor( 'C' , 'Power' , 100);
Rightie . TachoLimit = 80;
Rightie . SendToNXT();
Rightie . WaitFor();%Steuercode
Downp;
Down;
Down;
Right;
Right;
Up;
Up;
Uppie = NXTMotor( 'A' , 'Power' , -100);
Uppie . TachoLimit = 30;
Uppie . SendToNXT();
Uppie . WaitFor();
Left ;
Left ;
Leftieyo = NXTMotor( 'C' , 'Power' , 100);
Leftieyo . TachoLimit = 10;
Leftieyo . SendToNXT();
Leftieyo . WaitFor();
Upp;
Down;
Downp;
Right;
Right;
Upp;%Beispiel von Acht

```

Der folgende Code zeigt als Beispiel, wie die Zahl Acht geschrieben wird. Die darin enthaltenen Funktionen stellen die Bewegungsfunktionen der Kabine und des Auslegers dar.

Der Grund dafür, dass dieselbe Anweisung zweimal wiederholt werden muss, liegt an der Reibung zwischen dem Stift und dem Papier, und durch mehrere Versuche wurde herausgefunden, wo die Anweisung zweimal wiederholt werden muss.

LITERATURVERZEICHNIS

- [1] WIKIPEDIA: Regel von Sudoku ,<https://en.wikipedia.org/wiki/Sudoku>, Version:Februar 2024
- [2] Hans Andersson:
Sudoku Solver, <https://tiltedtwister.com/sudokudownload.html>, Veision:Februar 2024