Technische Hochschule
Brandenburg
University of
Applied Sciences
Fachbereich
Informatik und Medien

AMS-Projekt im WS 2023/24
Pizzabote

Kirill Kravets, Serafim Radkevitch

Projekt 5 Semester • Studiengang Informatik • Fachbereich Informatik und Medien • 11.01.2024

Aufgabenstellung

Ziel der Arbeit ist es, einen Roboter zu entwickeln, der eine oder mehrere Pizzen ausliefern kann. Das Streckennetz ist ein einfaches Gitternetz, bei dem es zu Störungen und zu unpassierbaren Kreuzungen kommen kann. Die gute Nachricht ist, dass man über das globale Wissen verfügt, um die aktuelle Karte der befahrbaren Wege dem Roboter kurz vor dem Start zur Verfügung stellen zu können.

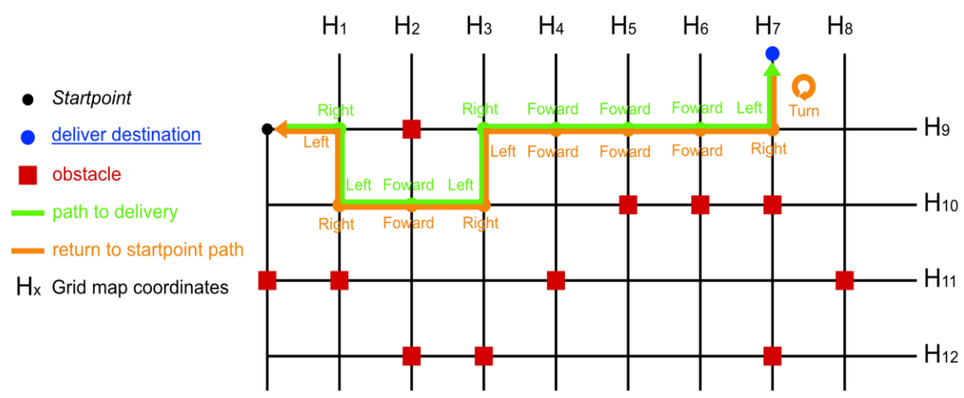

Abb. 1: Fahrplan des Roboters

Lösungsweg

Der Pizza-Bot war geplant, einem Staubsauger-Bot ähnlich zu werden, im Ziel, dass der Pizza-Bot flexibler fahren kann, doch die Idee hatte sich geändert, da es keine für den Kurs optimale Bau-Lösung war.

Das Design war geplant, sehr kompakt zu sein. Für den Roboter wurden zwei Räder implementiert, die pro Rad einen eigenen Motor haben, der asymmetrisch zu dem anderen Motor positioniert ist, damit der Roboter sich präzise auf einem Punkt umdrehen oder abbiegen kann, weswegen die Motoren mit den Zahnrädern für jedes Rad anders platziert wurden. Bei dem linken Rad ist der Motor im hinten Teil positioniert und bei dem rechten nach vorne (*Abb. 3*).

Da der Roboter nur zwei Räder hatte, hatten wir dem Gerüst vorne und hinten zwei kleine Lego-Säulen mit abgerundeten Bausteinen unten implementiert, um die Stabilität des Roboters zu verbessern. Die zweite Etage steht mithilfe von vier vertikal stehenden Legosteinen, was das Gewicht von dem Roboter verringern sollte, damit man nicht überflüssig viele Bausteine nur für das Tragen einer Plattform verschwendet.

An der vorderen Säule wurden zwei Optokoppler-Sensoren implementiert, da die Säule etwas breiter war, als die Linie, der der Roboter folgen sollte.

Für die Implementierung der Greif-Klau wurde ein Servomotor verwendet, der nicht zentral an dem Bot platziert wurde, weil es ausreicht, nur ein Zahnrad von nur einer Klauen-Hälfte anzudrehen, da die andere Hälfte sich immer mitbewegen wird. Der Auslöser-Knopf für die Klaue wurde an der vorderen Säule befestigt, da er die passende Höhe des Bestellung-Zielpunktes hat.

Hardware

Für die Hardware wurden vorne *2 Optokoppler (Abb. 2a)* verwendet, die das Linienfolgen ermöglichen. Auch da gibt es einen *Lichtsensor,* der für den Start verantwortlich ist. Drinnen im Roboter sind 2 *Motoren* (Abb. 3a) mit "1 zu 3" Getriebe (Abb. 3b). Jedes Rad kann sich in entgegengesetzte Richtungen drehen, was die Drehungen an den Kreuzungen präziser macht. Auf der höchsten Ebene des Roboters liegt ein *Servomotor (Abb. 2b)*, der den *Greifhaken* betreibt. Auch daneben liegen *Aksen-Board* und *Akkuträger (Abb. 2c)*.

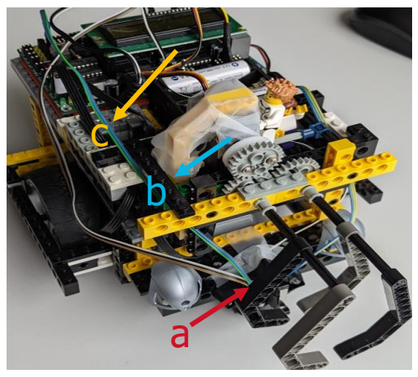

Abb. 2: Roboter

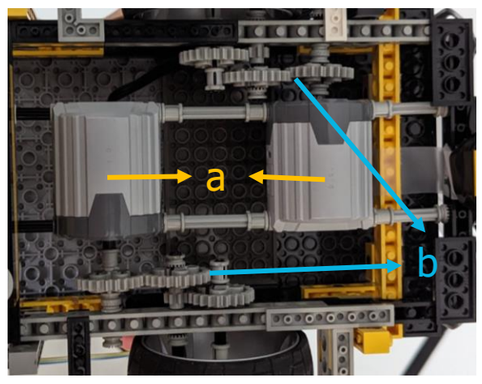

Abb. 3: Roboter (Sicht von unten)

Software

Unser Algorithmus nutzt Breitensuche - ein Verfahren in der Informatik zur Durchsuchung der Knoten eines Graphen. Die Breitensuche ist ein Algorithmus, der ausgehend vom Startknoten der Graphen die einzelnen Level der Graphen in der Breite durch Expansion nach einem Element durchsucht.

Am Anfang wird die Methode aufgerufen, die die Zeichenkette (Map) analysiert und uns den kürzesten Weg zum "D" ausgibt. Das erfolgt über die Analyse der benachbarten Schnittpunkte der Startposition, die in einem Array vorher gespeichert wurden *(Abb. 4a)*.

Falls "D" noch vorhanden und zugänglich ist, wird die Methode aufgerufen, die am Ende einen Weg von "S" zum "D" bildet *(Abb. 4b)*. Dieser Weg wird interpretiert, um ein Array mit den Anweisungen des Roboters zu erschaffen *(Abb. 4c)*. Diese Anweisungen sind: **L**eft, **R**ight, **U**_Turn und **F**orward. Nach der Lieferung wird ein umgekehrtes Array erzeugt und eine Route zurück wird erstellt *(Abb. 4d)*. Falls es keinen Weg mehr zum "D" gibt, wird eine 1 zurückgegeben, was zeigt, dass der Algorithmus beendet ist.

Auch gibt es einen eingebauten Timer, der den Roboter nach 120 Sekunden automatisch ausschaltet.

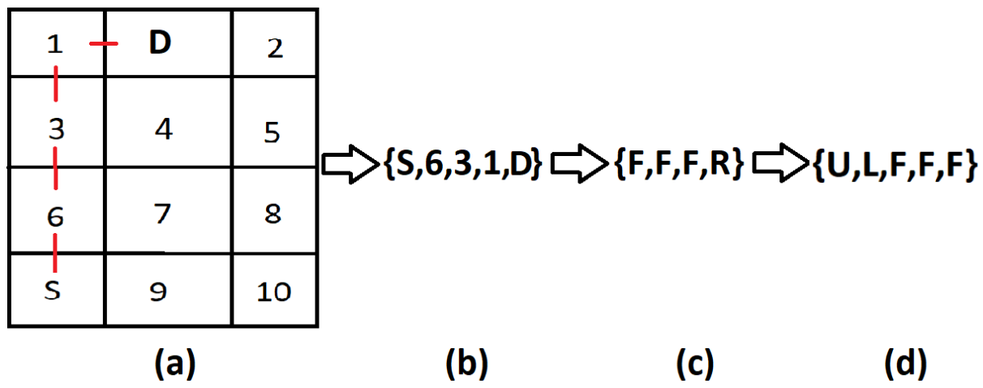

Abb. 4: Funktionsweise des Breitensuche-Algorithmus