

Bauversuch eines Erkundungsroboter

Mahmoud Jahjah, ETIT
Otto-von-Guericke-Universität Magdeburg

Zusammenfassung—Wie seit 2013 fand es auch dieses Jahr ein Lego Mindstorms Projektseminar an der Otto von Guericke Universität mit vielen praktischen und erfindereich Projekten. In dieses Paper geht es um einen Versuch, der sich mit der Entwicklung eines Roboters befasst. Der Roboter soll am Ende des Versuchs Weg von Anfangspunkt zum Endpunkt durch Hindernisse erkunden.

Schlagwörter—Wegsuch Algorithmus, A-Stern Algorithmus, Bug O Algorithmus, Roboter.

I. EINLEITUNG

MIT der Entwicklung der Zivilisation begann der Mensch zu suchen, was sein Leben in jeder Hinsicht einfacher machte. Um die Entfernung zu erleichtern, hat er in letzten 2 Jahrhunderten von der Dampfmaschinen bis des ersten Verbrennungsmotors und dessen Entwicklungen bis zu den Elektromotoren erfunden. Aber er war damit nicht zufrieden, sondern arbeitete er nun auch an Autos, um sich selbst zu führen. Heutige Fahrzeugtechnik interessiert sich dafür, dass die Maschine sich mit guten Verständnis der Umgebung im Raum bewegt. Obwohl diese Technologie Ihr endgültige Form noch nicht erreicht hat, bietet sie vielfältige Produkte: von Haushaltsprodukten wie elektronischen Staubsaugern bis zu selbst parkende Autos. In dem Projekt , das LEGO Mindstorms Seminars im Zeitraum von 2 Wochen mit Herr Altaraben und Herr Duwanoff bearbeitet wurde, wird ein Versuch durchgeführt, dass ein Roboter sich selbst von Startpunkt zu Endpunkt führt und die Hindernisse umgeht.

II. STAND DER TECHNIK

Der Roboter wird mit LEGO MindStorm NXT angeschaltet. Das NXT-Set ist doch besonderes, weil es NXT -Stein , Servmotoren und verschiedene Sensoren besitzt. Der NXT-Stein hat einen 32Bits-Mikroprozessor, Realsound-Lautsprecher und Anschlüsse für die Motoren und Sensoren, und lässt sich in MATLAB programmieren; Die Arten von Sensoren sind Ultraschallsensor, Lichtsensor, Schallsensor, und Farbsensor. Für den Roboter wird nur Ultraschallsensor benutzt, der den Abstand misst. Außerdem war die Aufbau des Roboters simpel, da die Programmierung der größte Teil des Projektes ist. und für die Bewegung haben wir uns A-Stern-Wegsuchung-Algorithmus entschieden.

III. HAUPTTILE

A. Aufbau und Program Struktur

Bei die Robotstruktur waren die beide Motoren mit den Vorderräder verbunden, und die hintere Räder durch Rädchen



Abbildung 1. Der Erkundungsroboter

mit die Vorderräder gekoppelt wurden. Für die Rechterotation des Robots werden die linke Rädchen mit Uhrzeige gedreht und sich die rechte gegen Uhrzeige gedreht, und vice versa, Außerdem wird Graphik-User-Interface(GUI) als die Control Panel des Robots gezeichnet. Mit zwei Optionen, entweder Manuel oder Autopilot. Im Autopilot wird die Koordinaten der Zielpunkt eingegeben. Außerdem die Leistung der Motoren wird auch vom User ausgewählt. Darüber hinaus wird ein Manuelkontroll addiert, dass der Roboter sich hin- und herbewegt und sich Links und Rechts orientiert.

B. Algorithmus

Warum haben wir A-Stern-Algorithmus festgestellt und wie funktioniert er? Was ist in dem besonderes? Er ermittelt den kostengünstigsten Weg zwischen einem Anfang- und einem Zielknoten innerhalb eines Suchbaums. Es könnte auch sein, dass der Suchbaum ein matrix-förmiger Suchbaum ist, weil bei der quadratischen Darstellung des Bewegungsfelds die Genauigkeit der Bewegung des Roboters in diesem Fall verbessert wird. Darüber hinaus ist er ein informiertes und erschöpfendes Verfahren, aber nur im Suchprobleme praktisch, im unseren Situation ist er sehr anwendbar. Bei Vergrößern der Knoten innerhalb des Suchbaums werden die Kosten aller neuen Knoten durch die Funktion :

$$K = K_b + H$$

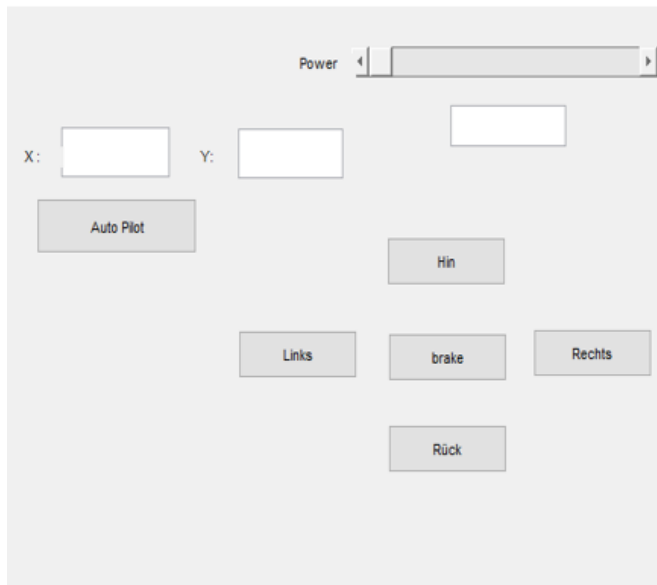


Abbildung 2. Graphic USer Interface (Control Panel)

Ermittelt, wo K Kosten des Knoten sind; K_b der Entfernung von dem Anfangspunkt ist; und H die Entfernung von dem Zielpunkt ist. Außerdem gibt es 2 Liste, eine heißt *OpenList* und die andere *CloseList*. Im *OpenList* werden die Kosten aller Knoten miteinander vergleicht. Der Punkt mit kleinsten Kosten wird im *CloseList* beseitigt. für bessere Erklärung wird eine Demonstration [1]

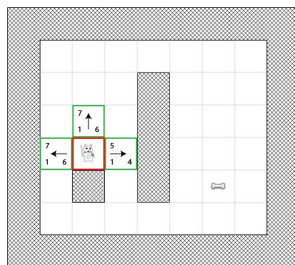


Abbildung 3. Anfangsbild

In diesem Beispiel wird das Rote Quadrat als Anfang und gegenwärtige Position der Katze bezeichnet. Außerdem wird der Knochen als Endpunkt gezeichnet. (siehe Figure 1) Jede grüne Quadrat stellt eine mögliche Bewegung der Katze dar. Zwei davon haben 7 als knoten Kosten K . Außerdem ist jede H werte rechts auf der unteren Ecke. Der Algorithmus geht dann zu die Kleinste K . in diesem Fall Rechts, und wird diese Quadrat als S genannt

Nachdem die Katze sich bewegt hat, wird S im Close Liste entfernt, und nochmal werden aller Kosten für die Punkte berechnet, aber dieses Mal haben wir ein Hindernis, d.h das die Katze kann nicht dadurch gehen. Hindernisse werden im Close Liste entfernt.(siehe Abbildung 3)

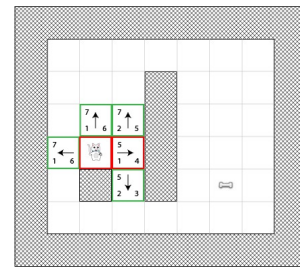


Abbildung 4. Die Katze hat sich bewegt und ein hindernisse sieht

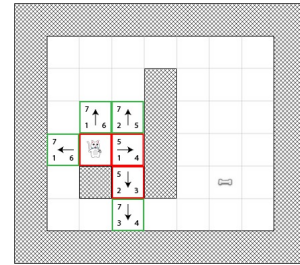


Abbildung 5. Sie hat die Hindernisse bemerkt und sich rechts bewegt

Nach der Entfernung des zweiten Punkts gibt es mehr als zwei Knoten, die gleiche K Werte haben. In diesem Fall wird die H Werte vergleicht. d.h das die Punkt mit Kleine H Werte wird bevorzugt.

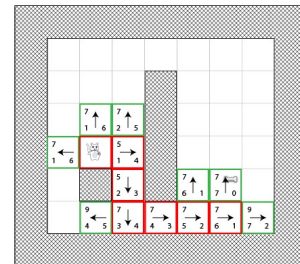


Abbildung 6. Die Katze hat das Ziel erreicht

Nach der Wiederholung des Vergleichens zwischen Knoten erreichtdie Katze ihr Zielpunkt (siehe Abbildung 6). Der Algorithmus ist, wie vorher gesagt wurde, ist nur im Such-probleme, aber in dem Projekt wird die Nutzung von A* mit Robotsbewegung geprüft. Der Algorithmus wird im Auto Pilot durchgeführt und mit manuelle Kontrolle im GUI angewendet.

IV. ERGEBNISDISKUSSION

A. Herausforderungen

Die algorithmus endgültige Form ist mit der Frist nicht erreicht geworden, deshalb wird der Algorithmus verändert.

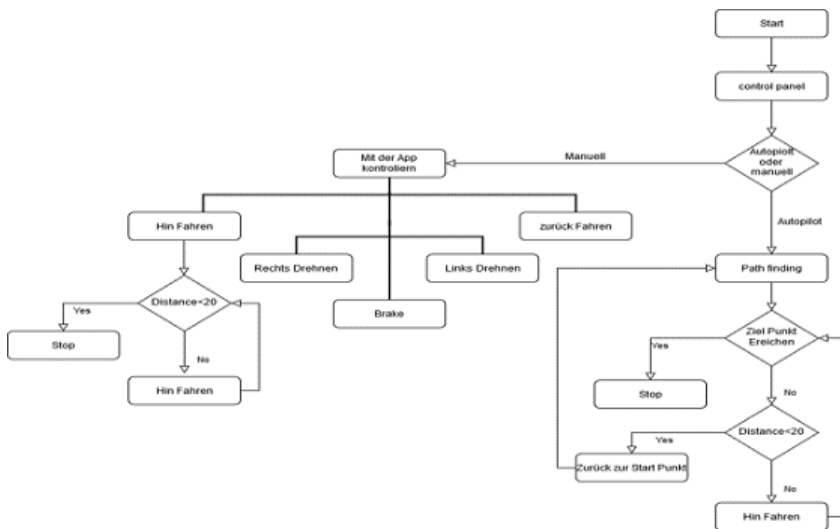


Abbildung 7. Ablaufplan

und wird ein anderer Algorithmus ausgewählt, der Bug O heißt. [2] Sein Hauptkonzept ist den direkten Pfaden zum Ziel, und wenn ein Hindernis siehst, folgt er es. Auf der anderen Seite treten aufgrund dieser Veränderung des Plans viele Behinderungen auf. z.B der Roboter die Hindernisse bis unendlich folgt. Außerdem wird die Genauigkeit des Roboterpositions mangelhaft, da die Abstand zwischen Anfang- und Endpunkt nicht punktlich ist. Beim Sturktur war die Ablenkung nicht immer getroffen, da die Rädchen nicht gut fest gehalten war.

B. Überwindungen

Die Verbindung von A* und Bug O Konzepten war die Lösung. Die unendliche Folgerung der Hindernisse und Positionsgenauigkeit kann durch die Anwendung der Matrix von A* übernommen werden, da es dem Roboter bessere Einsicht gibt. Die Anwendung des Bug O Konzepts ermöglicht, dass der Robot sich gerade und 45Grad bewegt.

V. ZUSAMMENFASSUNG

Wegen der Zeit war der Roboter nicht fertig, deshalb haben wir die Hauptkonzept von A* und anderen Algorithmus, Bug O, zusammen benutzt, damit Das Robot sich zum Ziel bewegt. Warum wird Bug O entschieden? Da er in wenigen Zeit programmiert werden kann.

LITERATURVERZEICHNIS

- [1] WENDERLICH, Ray: *Introduction to A* Pathfinding*. Version 03.00, September 2011. <https://www.raywenderlich.com/3016-introduction-to-a-pathfinding><https://www.raywenderlich.com/3016-introduction-to-a-pathfinding>. – PD 5214.5061.61
- [2] BLACK, Paul E.: *Big O notation*. 6. 2006