

Der Spotfinder

Daniel Behling, Elektromobilität
Otto-von-Guericke-Universität Magdeburg

Abstract- In dieser Arbeit handelt es sich um die Entwicklung und Realisierung eines mit der Software Matlab programmierten Roboters. Es war die Aufgabe, sich innerhalb von zwei Wochen mit dem Programm vertraut zu machen und in Zweiergruppen selbst eine Idee zu finden was der, mit Lego Mindstorms gebaute Roboter, einmal können soll. Die Arbeit unterteilt sich in eine Einleitung, bei der auf die Idee eingegangen und etwas zu den möglichen Anwendungsgebieten gesagt wird. In der darauf folgenden Vorbetrachtung werden dabei die bereits vorhandenen Lösungen vorgestellt. In dem Hauptteil wird auf das Konzept des Roboters und auf die Umsetzung eingegangen. Abschließend folgt dann noch eine kurze Ergebnisdiskussion und ein Fazit. Der „Spotfinder“ sollte in der Lage sein durch das Scannen des Untergrundes und dem Finden von farbigen Punkten selbstständig fortbewegen können. Je nachdem was dann für eine Farbe gefunden wurde, kann er zusätzlich festgelegte Befehle ausführen.

Der Spotfinder ist ein selbstständig fahrender Roboter der sich das Erkennen von Farben auf einem Untergrund fortbewegt.

Bei ihm handelt es sich nicht um einen Roboter, der im Voraus ein speziell festgelegtes Anwendungsgebiet hat. Vielmehr geht es dabei um die Funktions- und Herangehensweise des Roboters, was ihm ermöglicht ein breites Feld an Anwendungszwecken abzudecken. Das Anwendungsgebiet kann ihm also im Nachhinein zugewiesen werden.

Mögliche Gebiete wären die Untersuchung von Dellen, Kratzern oder Unebenheiten an Gebäuden, Brücken oder den Tragflächen eines Flugzeuges.

Wie der Name schon sagt soll der Spotfinder einen bestimmten Punkt finden, analysieren und danach auswerten.

Der Roboter sollte in der Lage sein, ein bestimmtes Gebiet bzw. einen bestimmten Untergrund selbstständig zu scannen und je nach Ergebnis verschiedene Schlüsse ziehen. Um dies zu ermöglichen muss er mit Sensoren ausgerüstet werden, die Veränderungen im Untergrund wahrnehmen können.

Wurde ein Unterschied wahrgenommen, muss der Roboter diese Information verarbeiten und auswerten. Und je nach Auswertung führt er dann einen ihm im Voraus einprogrammierten Befehl aus. Damit die Sensoren einen Bereich gleichmäßig abdecken können muss der Roboter zusätzlich kompakt und wendig sein.

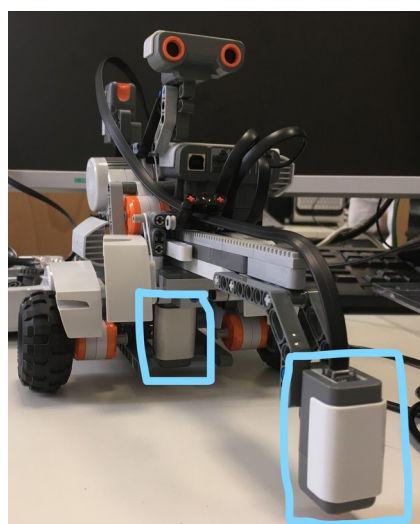


Bild 1:
Roboter mit
beiden
Sensoren

Um das Konzept der Spotfinders zu realisieren mussten ein paar Dinge beachtet werden. Als erstes musste geklärt werden wie der Roboter den Untergrund scannt. Dafür wurden zwei verschiedene Sensoren genutzt.

Der erste (vordere) sollte einen Unterschied im Untergrund erkennen und der zweite (hintere) dann genau feststellen um was für einen Untergrund es sich handelt.

Um einen Bereich gleichmäßig scannen zu können sollte sich der Roboter dafür um seine eigene Achse drehen. Ein drittes, loses Rad wurde für diesen Zweck angebracht um eine einfache Rotation zu ermöglichen.

Der Roboter sollte sich dann durch eine Endlosschleife so lange drehen bis er einen Wechsel auf der Oberfläche registriert hat. Nach dem registrieren stoppen die Motoren und er fährt so lange geradeaus bis der zweite Sensor den (im jetzigen Fall) Punkt gefunden hat.

Findet der erste Sensor nach 360 Grad drehen keinen Punkt fährt er seinen, mit einem Motor gesteuerten, Arm aus um so den Radius zu erhöhen.

Der Arm wurde durch ein Zahnrad ausgefahren und besaß drei verschiedene Stufen. Wenn er nach dem zweiten mal um seine Achse drehen erneut keinen gefunden hat würde er seinen Arm wieder ausfahren und so wiederholt sich der Prozess.

Ein Problem was dabei auftauchte war, dass sich die Vorrichtung durch ihr eigenes Gewicht nach unten gezogen hat und so der Abstand des Sensors zum Untergrund mit höherer Stufe immer geringer wurde.

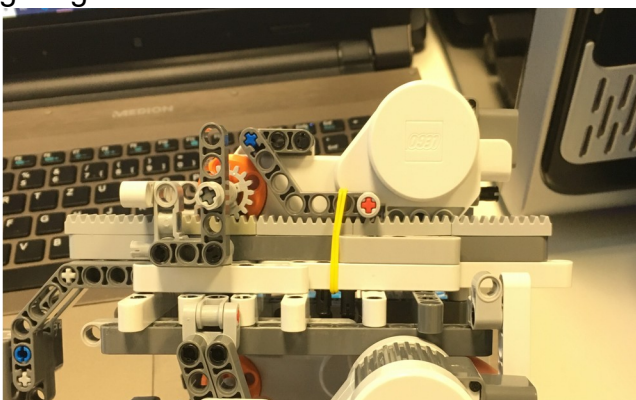


Bild 2: Vorrichtung des ausfahrbaren Armes

Dies führte wiederum zu Messfehlern des Sensors. Um das weites gehend zu verhindern wurde ein Gummiband eingespannt welches den Arm waagrecht hielt aber auch nicht zu viel Spannung besaß, sodass er jenen problemlos wieder einfahren konnte.

Das Scannen der Punkte erfolgte wie oben schon angerissen durch eine 360 Grad Drehung der Motoren, die dafür entgegengesetzt angesteuert wurden. Wenn jetzt bei den zweiten 360 Grad, also der ersten Stufe des Armes, ein Punkt registriert wurde, fährt er seinen Arm ein und so lange geradeaus bis der zweite Sensor den Punkt erkannt hat. Je nach dem, um welche Farbe es sich bei den Versuchen handelte, wurde entweder ein Ton abgespielt, die Entfernung nach vorne gemessen, die Raumhelligkeit ausgegeben oder im Programm Matlab ein Text angezeigt. Zusätzlich wurden dem Roboter zwei Abbruchkriterien einprogrammiert. Das erste war eine bestimmte Farbe z.B. schwarz, die vorab vom Nutzer festgelegt wurde. Das zweite Kriterium war, dass wenn der Sensor alle drei Radien gescannt hat und trotzdem keinen Punkt finden konnte.

Für den Fall, dass eines der beiden Kriterien eintraf, sollte der Roboter wieder zum Startpunkt fahren.

Um dies zu realisieren, wurde jeder Winkel und jede Entfernung der einzelnen Punkte abgespeichert und die Motoren nach jedem scann des zweiten Sensors auf Null zurückgesetzt.

Die Werte wurden in einer Matrix abgespeichert und Rückwärts abgerufen, sodass der Roboter in der Lage war seinen abgefahrenen Weg wieder zurückfahren und sich in seine Ausgangsposition begibt.

Abschließend kann man sagen das der Roboter am Ende in der Lage war, alle ihm gestellten Anforderungen im Grunde bewältigen zu können. Er hat mehrere Tests problemlos durchlaufen können und zum Ende nur noch selten Fehlermeldungen ausgegeben. Jedoch sind uns auch während den ersten Tests immer wieder kleine Fehler und Verbesserungsmöglichkeiten offensichtlich geworden.

Ein Problem, was während der Testphase bemerkbar wurde, ist das der Spotfinder mit seinem vorderen Sensor zwangsläufig den vorherigen Punkt, der ja 180 Grad hinter ihm lag, überfahren und wahrnehmen würde. Um dieses Problem zu lösen wurde ihm einprogrammiert alle Punkte in einem bestimmten Bereich hinter ihm (ca. 175- 195 Grad) zu ignorieren.

Dies wurde dann auch wie erwartet ausgeführt und bereitete dann keine Schwierigkeiten mehr.

Ein Fehler, welcher jedoch nicht so leicht durch das programmieren lösbar war, ist die Ungenauigkeit und Empfindlichkeit der Lego Mindstorms Sensoren selber. Tests die am Vormittag noch reibungslos funktioniert haben, weil die Sonne noch nicht geschienen hat, wiesen ab Mittag fast immer Fehler auf. Zum Beispiel wurden gelbe Punkte als weiß erkannt und blau als schwarz.

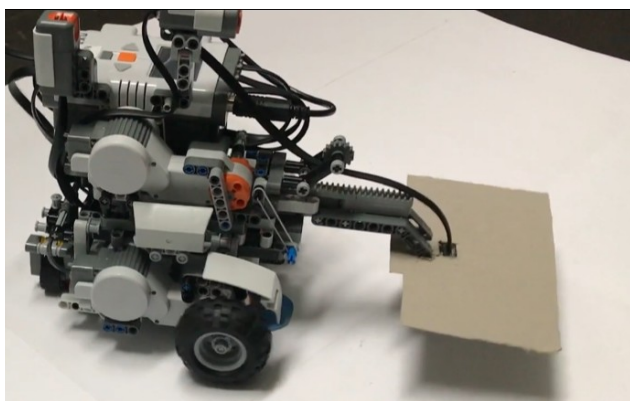


Bild 3: Spotfinder mit Verdeck am vorderen Sensor

Eine relativ simple und einfache Lösung dafür war ein Verdeck aus Pappe, die für einen gleichmäßigen Lichteinfall sorgte. Eine weitere Maßnahme die bei den Tests sehr geholfen die Messergebnisse zu verbessern war die Verwendung von möglichst hellen und nicht reflektierenden Farben.

Ein weitere Komplikation die während Versuchen deutlich wurde, ist die Prozessorleistung des NXT Computers selber. Als probiert wurde einige Prozesse bei den Tests zu beschleunigen um alles möglichst flüssig aussehen zu lassen, wie z.B. den Roboter nach dem scannen des schwarzen Punktes schneller zum Startpunkt zurück zu navigieren war nicht möglich.

Der NXT konnte die Masse an Informationen nicht so schnell verarbeiten und gab Fehlermeldungen aus. Deshalb wurden nach bestimmten Bewegungen immer kurze Pausen einprogrammiert, um den Spotfinder die Zeit zu geben seine Aufgaben ohne Probleme auszuführen.

Insgesamt gab es aber keine Probleme die nicht behoben werden konnten. Und als Ergebnis ist zu sagen das der Roboter allen vorab gestellten Erwartungen gerecht wurde.

Das Fazit für den Roboter als auch die zwei Projektwochen ist durchweg positiv. Nach nur zwei Wochen war man in der Lage einen eigenen Roboter zu entwerfen und mit Hilfe der Software Matlab zu programmieren. In dem Fall ein Roboter der sich auf einer Fläche mittels farbiger Punkte, selbstständig fortbewegen konnte und in der Lage war seinen abgefahrenen Weg wieder zurückzulegen. Für zukünftige Verbesserungen/ Optimierungen könnte man mehr Suchradien einführen oder den Mechanismus komplett erneuern, sodass sich der Spotfinder dauerhaft dreht und dabei seinen Arm ein- und ausfährt.

Zusätzlich könnte man noch ein GUI (guided user interface) erstellen, mit welchem der Benutzer selber und immer kurz vor dem durchlaufen des Programms einer Farbe einen jeweiligen Befehl zuordnen kann, um dies umzusetzen waren zwei Wochen jedoch zu kurz. Ein persönlicher Verbesserungsvorschlag wäre vielleicht eine dritte Woche hinzuzufügen, da man allein eine Woche braucht um sich mit Matlab vertraut zu machen und sich so noch viel mehr Möglichkeiten bieten. Abgesehen davon war es eine sehr schöne, interessante, aber auch anspruchsvolle Erfahrung, die es jedoch absolut Wert war.