

Fußballroboter – Ein LEGOMindstorms Projekt

Fabio Näther, Studiengang
Otto-von-Guericke-Universität Magdeburg

Mithilfe der LEGO Mindstorms Baureihe sollte während des Praktikums ein Roboter oder Anlage gebaut werden, welche in der Lage sein sollte, eine selbstformulierte Aufgabe eigenständig zu bewältigen. Verfügbar waren dafür ein programmierbarer Legosteine NXT sowie verschiedene Sensoren und Motoren. Ebenfalls gehörten LEGO Technik und normale LEGO-Bausteine zu den verbaubaren Teilen. Limitierende Grenzen waren dabei durch die maximale Anzahl an Motoren, der NXT verfügt über 3 Motoranschlüsse, und die maximale Anzahl an Sensoren, nämlich 4 Sensoranschlüsse gegeben. Die Programme wurden dabei über die Programmiersoftware MATLAB erstellt und anschließend über USB-Kabel oder Bluetooth übertragen.

I. EIGENE AUFGABENSTELLUNG

Ziel des Roboters war es einen Ball innerhalb eines begrenzten Spielfeldes zu erkennen und eigenständig in ein Tor zu bewegen. Um diese Aufgabe lösen zu können, musste der Roboter in der Lage sein, seine eigene Position im Spielfeld als auch die Position des Balls zu erkennen. Er musste außerdem den Ball sicher aufnehmen und wieder ablegen können. Zudem musste er in der Lage sein, den Ball in das Tor zu bewegen. Um diese Aufgaben lösen zu können, musste der Roboter einen Antrieb besitzen, durch den er das Spielfeld befahren konnte. Dafür war es notwendig, eine Steuerung für eben jenen Antrieb zu programmieren. Für die Aufnahme des Balls war es erforderlich, dass der Roboter sich zum Ball bewegt und diesen greift. Um das Tor zu erzielen, musste der Roboter in der Lage sein, sich zum Tor auszurichten und den Ball in der richtigen Position abzulegen.

II. BAU DES ROBOTERS

A. Bau des fahrbaren Chassis

Um die Aufgabe des Antriebes zu bewältigen, wurde entschieden, ein Kettenfahrzeug zu bauen und dafür 2 der 3 möglichen Motoren zu benutzen. Nach Tests mit Ketten- und Radfahrzeugen stellte sich heraus, dass eine genaue Drehung des Fahrzeuges nur möglich wäre, wenn das Radfahrzeug über eine Lenkung verfügen würde. Das einfache Abstellen eines Motors bei einem Radfahrzeug führte zu großen und ungenauen Drehungen. Da dies jedoch ein Hauptbestandteil der Erfassung des Balls werden würde, war es nur mithilfe des dritten Motors möglich, eine solche Genauigkeit zu erzeugen. Da jedoch

dieser Motor bereits für das Öffnen und Schließen des Greifers benötigt wurde, war ein Radfahrzeug keine Option. Aufgrund dieser Probleme wurde sich für ein Kettenfahrzeug entschieden, welches 2 Ketten besaß. Dabei wurde jeder dieser Ketten von einem eigenen der beiden für diese Aufgabe abgestellten Motoren bewegt. Diese wurden direkt durch eine Achse mit dem Laufrad der jeweiligen Kette verbunden. Beide Motoren wurden miteinander verbunden, um die nötige Stabilität zu erzeugen. Dies geschah durch verschieden längliche LEGO Steine und Stangen. Auf den beiden Motoren, welche die Ketten antrieben wurde der NXT befestigt und die beiden Motoren wurden durch Kabel an den Motoranschlüssen des NXT angeschlossen. Diese Anordnung ermöglichte es durch das Drehen einer Kette in Fahrtrichtung und eine in die Entgegengesetzte, eine perfekte Drehung auf der Stelle durchzuführen. Nach einigen Test mit dieser Konfiguration stellte sich heraus, dass die beiden Motoren für eine Drehung auf der Stelle nicht nur entgegengesetzte Richtungen sondern auch mit verschiedenen Geschwindigkeiten und Laufzeiten aktiviert werden mussten. Das Verhältnis dieser Faktoren wurde in einigen Test herausgefunden und stellte somit eine genaue Drehung sicher, welche für spätere Aufgaben unabdingbar war.

B. Bau des Greifers und der Ballerkennung

Um die Aufgabe des Greifen und Ablassen des Balls zu ermöglichen, war der Bau eines Greifers nötig. Dieser sollte in der Lage sein den Ball sicher zu erkennen und zu greifen. Außerdem war es nötig in anschließend wieder sicher abzulassen. Die Bewegung dieses Greifers übernahm der dritte Motor, welcher liegend über dem NXT befestigt wurde. Durch eine Achse wurde eine der in Fahrtrichtung linke Greiferarm direkt angetrieben. Der zweite Greiferarm wurde durch drei in Reihe liegende Zahnräder bewerkstelligt. Durch das Verwenden von einer ungeraden Anzahl an Zahnräder wurde es möglich die beiden Greiferarme mit nur einem Motor zu öffnen und zu schließen. Auch dies war entscheidend, da nur ein Motor zur Verfügung stand. Der Motor wurde über ein Kabel am dritten Motoranschluss am NXT angeschlossen. Der Ball sollte mit drei senkrecht nach unten gerichteten Farbsensoren erkannt werden. Ziel dabei war es, aus der Bewegung heraus den Ball zu erkennen und anzuhalten um den Ball zu greifen. Die drei Farbsensoren wurden gleichmäßig über der Spannweite des Greifers verteilt und an die ersten drei Sensorenanschlüsse am NXT durch Kabel angeschlossen. Eben jene begannen bei Fahrtbeginn den Boden unter ihnen zu scannen und die

Farbe zu speichern. Dies taten sie alle 20 ms. Der Roboter bewegte sich bei diesem Teil des Programms innerhalb einer Schleife welche die Farbwerte der Sensoren prüfte. Sollte einer der Sensoren einen roten Wert aufnehmen

würde die Schleife beendet werden und der Roboter stoppen. Anschließend sollte der Teil des Programms ablaufen, welcher den Greifer sich schließen lies. Nach einigen Test wurde das Problem deutlich, dass der Stopp des Roboters verzögert sein würde und dadurch der Ball berührt werden würde. Dies führte dazu, dass der Ball in Bewegung gebracht wurde und nicht mehr vom sich schließenden Geifer gesichert werden konnte. Für die Lösung dieses Problems stoppte der Roboter beim Erkennen eines roten Objektes nicht sofort ab sondern bewegte sich in langsam Tempo weiter gerade aus. Dadurch wurde der sich vom Roboter entfernende Ball gesichert.

C. Erkennen des Roboters und des Balls

Um den Roboter und den Ball auf dem Spielfeld zu erkennen bot sich die Nutzung einer Webcam an. Diese wurde über dem Spielfeld platziert und fotografierte das Spielfeld aus einer Höhe von circa 1.70 m. Idee dabei war es das gesamte Spielfeld in kleine Quadrate aufzuteilen. Diese würden das Abmaß der Auflösung der Kamera haben. Das Programm sollte dabei jedes einzelne Quadrat nach dessen Farbzusammensetzung überprüfen. Dafür wurden die RGB Werte in drei, der Auflösung der Kamera entsprechende, Tabellen eingetragen. Da unser Ball rot war, hat er alle von ihm eingenommenen Felder nur mit einem rot Wert beschrieben. Mithilfe einer Gleichung wurden von der Kamera fehlerhaft, jedoch immer nur mit kleinen rot Werten, versehene Felder ignoriert. Hatte das Programm die Werte erhalten suchte es nach Werten, welche es berücksichtigen sollte und rechnete den Mittelpunkt dieser Felder aus. Der Wert dieses Feldes in der Tabelle, also sein Spalten- und Zeilenwert wurden dabei an zwei Variablen übergeben. Somit war die Position des Balls im Spielfeld erkannt. Um die Position des Roboters zu erkennen wurde dieser mit einem grünen und einem blauen Punkt versehen. Der grüne Punkt wurde dabei direkt über dem Drehpunkt des Fahrzeuges platziert (Erklärung folgt). Der blaue Punkt wurde in der Längsachse des Fahrzeuges in Fahrtrichtung vor dem grünen Punkt platziert. Vom selben Teil des Programms wurden jeweils zwei weiteren Variablen die Werte des grünen und des blauen Punktes übergeben.

D. Ausrichtung des Roboters zum Ball

Um nach der Erkennung des Balls und des Roboters diesen Auszurichten wurden die in Punkt C. gespeicherten Variablen verwendet. Idee war es durch das Berechnen des Winkel des Roboters und es Roboters zum Ball einen Winkel zu berechnen, welcher verwendet werden konnte um sich zum Ball zu drehen. Dafür wurde

zuerst der Winkel des Roboters berechnet. Dies geschah mithilfe der atan2 Funktion und den gespeicherten Variablen. Da Programm rechnete einen Winkel zwischen dem grüne und dem blauen Punkt aus. Dieser lag im Wertebereich zwischen 180° und -180° . Anschließend wurde der Winkel zwischen dem grünen Punkt und dem Ball mit der gleichen Funktion berechnet. Die auszuführende Drehung des Roboters wurde anschließend durch die Subtraktion des Roboter Winkel vom Winkel zum Ball. Der resultierende Winkel war der zu drehende Winkel. Durch den Wertebereich von 180° bis -180° war es möglich eine zu drehende Richtung anzugeben. Mithilfe einer If-Else-Schleife entschied das Programm nach links oder nach recht zu drehen. War der Winkel größer als 0 drehte der Roboter sich nach links. War der Winkel negativ drehte er sich nach links. Der grüne Punkt wurde dabei exakt über dem Drehpunkt platziert um das berechnen des Winkels zu vereinfachen und die anschließende Drehung genauer zu gestalten. Dadurch wurde ein am Ball vorbei fahren, verhindert.

E. Ausrichtung des Roboters zum Abschlusspunkt

Um den Roboter in eine Position zu bringen, welche sich vor dem Tor befindet, wurde das in Punkt D. erläuterte Verfahren erneut angewandt. Der einzige Unterschied hierbei war, das ein neues Bild und Positionsvariablen erstellt wurden. Dies geschah wie in Punkt C. Beschrieben. Die Werte des Balls wurden dabei ersetzt durch einen festgelegten Abschlusspunkt ('Elfmeterpunkt'). Nachdem der zu drehende Winkel berechnet war wurde die Fahrtstrecke bestimmt. Dafür wurde ein Vektor vom grünen Punkt zum festgelegten Punkt berechnet. Anschließend errechnete das Programm die Länge dieses Vektors und gab dem Roboter die Anweisung die zu fahrende Strecke zurückzulegen. Nach dem sich die Motoren dich notwendige Strecke gedreht hatten blieb der Roboter stehen.

F. Ausrichtung des Roboters zum Tor

Das gleiche in E. erläuterte Verfahren wurde erneut angewandt. Einziger Unterschied hierbei war, dass der Abschlusspunkt mit seinen Variablen durch einen Torpunkt ersetzt wurde. Auch bewegt sich der Roboter nach dem Ausrichten zum Punkt nicht auf ihn zu.

G. Torschuss

Der Roboter öffnet den Greifer und fährt eine gewisse Strecke rückwärts. Anschließend fährt er mit maximaler Geschwindigkeit ein wenig weiter als der Anlauf gewesen ist und trifft damit den Ball. Der Ball bewegt sich durch das Anstoßen durch den Roboter in Richtung des Tores.

III. UMSETZUNG DES PROJEKTES

Die im Bau des Roboters beschriebenen Punkte fanden bei der Umsetzung der Idee nacheinander statt. Dabei traten immer wieder Probleme auf, welche teilweise durch das anpassen einzelner Werte und erneutes Testen aber auch durch komplettes abändern von Abläufen, behoben wurden. Durch das Erstellen einzelner Unterprogramme war es immer wieder möglich einzelne Teile des Programms zu testen und zu verbessern. Nachdem die einzelnen Unterprogramme funktionierten und in einem einzelnen Programm zusammengefasst waren, liefen sie wie in folgendem PAP zu sehen, ab.

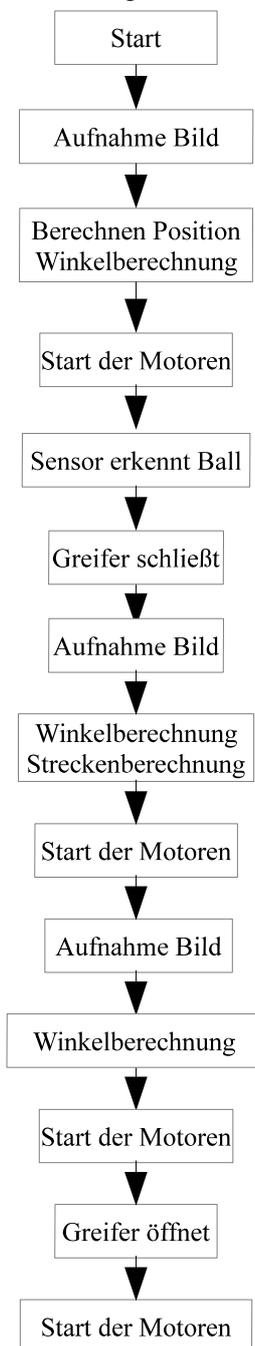


Abbildung 1: Programmablaufplan

Nach dem Start des Programms wird es zuerst über Bluetooth an den NXT übertragen. Nachdem dies abgeschlossen ist wird das erste Bild mit der Webcam geschossen. Nachdem dies geschehen ist setzt das Programm bei dem in Punkt C. geschilderten Ablauf ein. Dabei speichert es die entsprechenden Variablen und geht in den Punkt D. Über. Hat sich der Roboter ausgerichtet beginnt er sich mit halber Geschwindigkeit nach vorne zu bewegen. Und beginnt mit der in Punkt B. beschriebenen Suche nach dem Ball. Ist dies geschehen wird erneut ein Bild geschossen und der Roboter beginnt mit dem in Punkt E. beschriebenen Programm. Das Programm überschreibt dabei die noch vorhandenen alten Werte der Variablen. Es wird danach erneut ein Bild geschossen und der Punkt F. Wird ausgeführt. Anschließend folgt der finale Teil mit dem Punkt G..

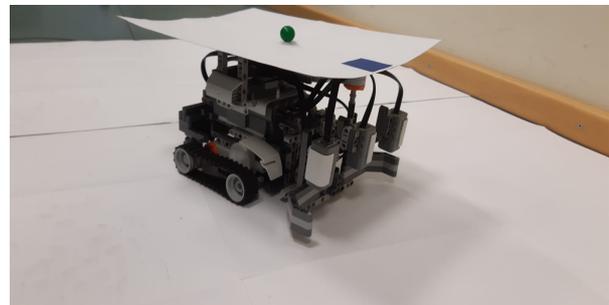


Abbildung 2: Aufbau Roboter mit Punkten

In Abbildung 2 ist der fertige Roboter mit dem grünen Punkt und dem blauen Punkt zusehen. Die angebrachte weiße Pappe soll dabei Farbige stellen auf einzelnen Bauteilen des Roboters verdecken und so die Positionsbestimmung genauer machen. Im vorderen Bereich des Roboters sind die beiden Greifarme und die drei Farbsensoren zu erkennen.

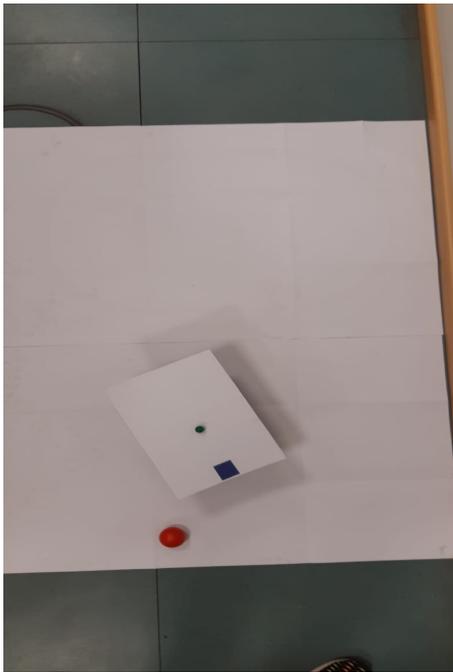


Abbildung 3: Sicht der Kamera

In der Abbildung 3 ist ein aufgenommenes Bild der Kamera zu sehen. Anhand eines solchen Bildes berechnet das Programm die Mittelpunkte des grünen und blauen Punktes und des Balls. Und übernimmt diese Werte als Variablen für die späterer Winkelberechnung.

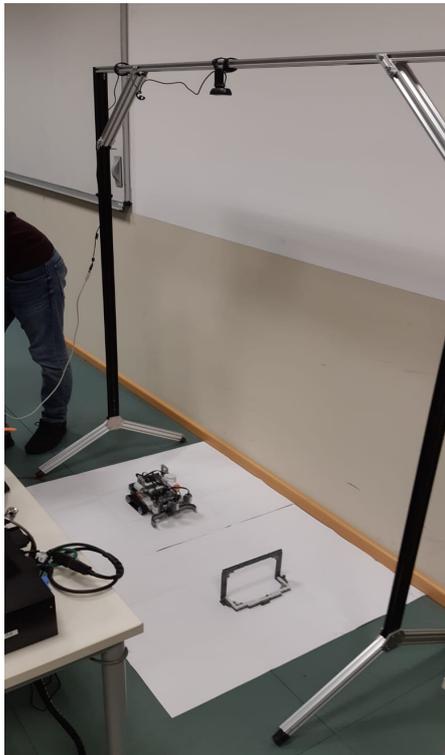


Abbildung 4: Gesamter Aufbau

In Abbildung 4 ist der gesamte Aufbau zu sehen. Die Kamera ist am zu sehenden Stativ befestigt, welches das gesamte Spielfeld überspannt. Sie ist über ein USB-Kabel mit dem Computer auf, welchem das Programm läuft,

verbunden. Es ist außerdem der Roboter und das Tor zu sehen. Das Spielfeld ist hierbei in etwa durch die weißen Planen markiert.

IV. AUSWERTUNG DES PROJEKTES

Als Ergebnis des Projektes ist ein Roboter zu betrachte, welcher selbständig in der Lage ist seine Position und die eines Ball in einem Spielfeld zu erkennen. Sich zu diesem auszurichten und ihn aufzunehmen. Sich anschließend zu einem Tor auszurichten und dieses im Anschluss zu treffen. Zudem gehören zu diesem Projekt das Tor und die notwendige Kamera. Beim Bau dieses Roboters traten verschiedene Probleme auf. Die genaue Steuerung des Chassis erwies sich als anfängliches Problem. Zudem gestaltete sich das Greifen des Balls mit dem Greifer als Problem, welches zu großen teilen gelöst werden konnte. Die Winkelberechnung wurde durch die Verwendung der atan2-Funktion im Vergleich zur davor verwendeten Berechnung durch zwei Vektoren deutlich verbessert. Die genaue Ansteuerung des Abschlusspunkt gestaltet sich schwierig da eine genaue Längeneinschätzung mit den Motoren nur schwer möglich ist. Ein weiteres Problem tritt auf wenn der Ball nah am Rand des Spielfeld liegt. Durch den Ablauf des Programms dem Ball nach erkennen noch ein Stück hinterher zu fahren kann es passieren, dass beim folgenden Bild Teile des Roboters mit den Erkennungspunkten nicht mehr im Spielfeld stehen.

V. ZUSAMMENFASSUNG UND FAZIT

Innerhalb des Praktikums entstand ein Roboter, welcher in der Lage ist ein Tor eigenständig zu erzielen. Dafür wurden grundlegende Programmierabläufe in MATLAB und die Funktion von LEGO Mindstorms erlernt. Außerdem wurden mehrere Vorträge gehalten und diese Facharbeit verfasst. Für das Projekt fehlend sind noch genauere Verbesserung des Roboters um die Trefferwahrscheinlichkeit zu erhöhen. Zudem wäre es vorstellbar Hindernisse zu erkennen und zu um fahren. Auch wäre es möglich, sich den Weg des Roboters in einem Koordinatensystem zur besseren Auswertung auszugeben.

VI. QUELLEN

Für dieses Projekt benutzen wird die Matlab Hilfsfunktion als auch den sehr nützlichen Rat unserer Betreuer.