

Tic-Tac-Toe Roboter

Svenja Langer, Elektro- und Informationstechnik
Otto-von-Guericke-Universität Magdeburg

Abstract — Im Folgenden wird sich mit dem Thema eines interaktiven Roboters befasst, der in der Lage ist, gegen einen menschlichen Mitspieler Tic-Tac-Toe zu spielen. Dabei wird auf den Aufbau und die Konstruktion eingegangen, die den Roboter befähigen, mittels eines Stiftes das Feld und seine Spielzüge zu zeichnen. Des Weiteren wird auf die Programmierung und den erstellten Spielalgorithmus eingegangen. Diese analysieren das Spielfeld und berechnen die Züge des Roboters auf Grundlage der aufgenommenen Daten. Die Aufnahme des Spielfeldes erfolgt dabei mit Hilfe einer Kamera über dem Spielfeld. Der Roboter ist in der Lage, eine vollständige Runde zu spielen, unabhängig davon, ob es einen Gewinner gibt oder das Spiel mit einem Unentschieden abgeschlossen wird.

Schlagwörter — Interaktiv, LEGO Mindstorms, Roboter

I. EINLEITUNG

Die Bedeutung von Robotern nimmt in der modernen Zeit immer mehr zu. Sie kommen in den verschiedensten Bereichen zum Einsatz und übernehmen teilweise die Aufgaben der Menschen und sollen sie somit entlasten. So gibt es mittlerweile auch die verschiedensten Spielroboter, die es ermöglichen, dass bestimmte Spiele auch ohne zweiten menschlichen Spieler gespielt werden können. Ein Beispiel für solche Roboter sind Schachroboter. Sie ermöglichen es, die Spielregeln und Strategien zu erlernen und zu üben. So kann unabhängig von einer weiteren Person zu jeder Zeit gespielt und gelernt werden. Das Prinzip des Spielens gegen einen nicht menschlichen Gegner könnte auch auf schwierigere Spiele oder Projekte zum Zweck des selbständigen Erlernens übertragen werden.

Ein weiteres einfacheres Beispiel für einen interaktiven Roboter wäre der Tic-Tac-Toe Roboter, welcher das Ziel in der Projektwoche war.

Der Roboter sollte zwei grundlegende Aufgaben bewältigen können.

Zum einen sollte er das typische Spielfeld selber zeichnen und innerhalb des Feldes die einzelnen Kästchen gezielt ansteuern. Dazu gehörte auch, dass der Roboter in das angesteuerte Feld sein Zeichen als Markierung setzt.

Die zweite Aufgabe bestand darin, tatsächlich spielen zu können. Dazu musste der Roboter erkennen, welche Züge der menschliche Mitspieler macht und musste dementsprechend mit eigenen Spielzügen reagieren. Wichtig dabei war, dass der Roboter nicht zufällig leere Felder füllt, sondern gezielt versuchte, selber zu gewinnen. Wenn das Beenden der

Spielrunde mit einem Sieg des Roboters nicht möglich war, sollte der Sieg des Gegners gezielt verhindert werden.

Der Aufbau des Roboters wurde mit LEGO-Bauteilen sowie der NXT-Station realisiert. Die Programmierung erfolgte mittels MATLAB und gewährleistete, unter Verwendung des Toolkits der RWTH Aachens, die Ansteuerung des NXT und somit auch der Motoren. Zusätzlich wurde eine Webcam zur Spielfelderfassung verwendet.

II. VORBETRACHTUNGEN

Der geplante Tic-Tac-Toe-Roboter basiert im Grunde auf einem Schreibroboter. Er ist geplant als eine Art Schreibmaschine, die anstatt verschiedener Buchstaben, das Spielfeld und die Züge zu Papier bringt. Eine gute Vorlage und Orientierungshilfe liefert der „LEGO Printer“.

Diese Schreibmaschine muss zusätzlich mit einem Spielalgorithmus sowie einer Spielfeldererkennung ausgestattet werden. Für die Erkennung des Feldes gibt es zwei Möglichkeiten, die in bisherigen Robotern verwendet wurden.

A. LEGO Printer

Die „LEGO Telegraph Machine and Printer“ von J. K. Brickworks [1] ist eine aus LEGO-Bauteilen und NXT-Einheit bestehende Schreibmaschine, die Morsesignale als Buchstabentext ausgeben soll. Dabei werden die Morsesignale über einen Taster erfasst, vom Programm ausgewertet und dann per Stift auf ein Blatt Papier gebracht. Der Roboter ist so gebaut, dass der angebrachte Stift in zwei Richtungen bewegt werden kann. Einmal kann der Stift von rechts nach links und dann noch hoch und runter bewegt werden. Die Auf- und Abbewegung wird durch das Kippen der gesamten Bewegungsvorrichtung des Stifts realisiert. Für jede der beiden Richtungen wird je ein Motor verwendet. Über einen weiteren Motor kann die Position des Blattes verändert werden. Mit dem Bewegen des eingelegten Blattes wird ermöglicht, dass tatsächlich Buchstaben statt horizontalen Linien auf dem Blatt entstehen.

Ein weiterer Sensor ist unterhalb des Roboters angebracht. Der Farbsensor soll die Erkennung der Blattränder oben und unten übernehmen.

B. Möglichkeiten der Spielfeldererkennung

Einerseits gibt es die Möglichkeit mittels der Licht- bzw. Farbsensoren von LEGO das gesamte Spielfeld abzufahren und somit Farbveränderungen innerhalb eines Feldes zu registrieren und an den Roboter zur Verarbeitung weiterzugeben [2].

Die andere Möglichkeit wäre die Verwendung einer Kamera,

welche das Spielfeld aufnimmt und die Veränderungen berechnen kann [3].

III. REALISIERUNG

A. Aufbau

Der Tic-Tac-Toe-Roboter ist aus LEGO-Bauteilen und einer NXT-Einheit, welche die drei verbauten Motoren ansteuert, aufgebaut.

Wie auch bei dem Roboter von J. K. Brickworks muss der Tic-Tac-Toe-Roboter einen Stift bewegen können, daher wurde die Umsetzung für die Bewegung von links nach rechts teilweise übernommen. Sie wird mit einem der Motoren angetrieben.

Um nicht nur durchgehende Linien zeichnen zu können, sorgt ein weiterer Motor dafür, dass der Stift angehoben und gesenkt werden kann. Um dies zu ermöglichen, ist der Stift mit Gummibändern befestigt. Zusätzlichen Halt bekommt der Stift durch eine angeklebte Zahnradschiene, die in ein Zahnrad greift und das Senken und Heben des Stiftes, im Gegensatz zur ersten Variante, kontrollierbar macht. Bei dieser ist die Schiene mittels weiterer Gummis am Stift befestigt. Dadurch können die Bewegungen nicht präzise durchgeführt werden und die Schiene wird eher entlang des Stiftes verschoben anstatt ihn tatsächlich zu bewegen. Der verwendete Motor, der direkt mit dem Zahnrad verbunden ist, ist so verbaut, dass er den Links- und Rechtsbewegungen des Stiftes folgen kann. Der dritte Motor treibt die Hinterachse an und ermöglicht es dem Roboter, sich vor und zurück zu bewegen. Die Vorderräder werden durch den Schub der Hinterachse bewegt. Für genaueres Anfahren des Spielfeldes treibt der Motor die Hinterachse nicht direkt, sondern über Zahnradgetriebe mit einer 9:1-Übersetzung an. Der gesamte Roboter fährt auf Rädern, statt ursprünglich geplant auf Schienen. Die Lage der verwendeten Motoren ist in der Abbildung 1 erkennbar.

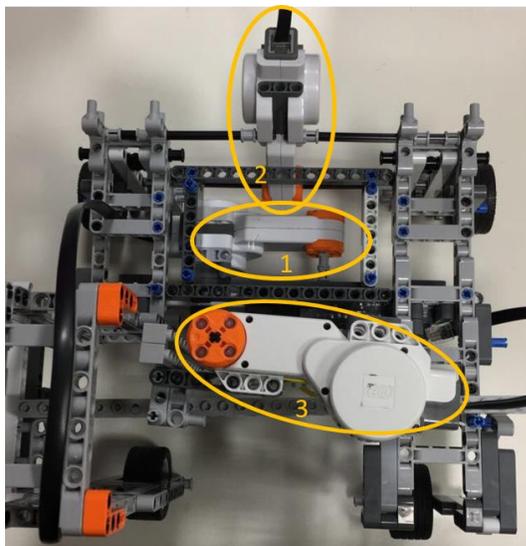


Abbildung 1: Roboter von oben
 1 – Motor für links und rechts
 2 – Motor für vor und zurück
 3 – Motor für hoch und runter

Als Untergrund für den Roboter wird eine große LEGO-Platte verwendet. Auf der Platte ist ein Gestell befestigt, welches als Kamerahaltung fungiert. Die Kamera hat somit nach dem Einhängen immer einen konstanten Abstand zum Spielfeld und das Feld liegt immer im gleichen Bildbereich. Die Verwendung der Kamera zur Felderkennung und Auswertung wurde gewählt, weil es im Vergleich zur Variante mit dem Sensor weniger Zeit beansprucht, das Feld zu analysieren. Um dagegen mit einem Sensor das Feld zu erfassen, müsste der Roboter das Feld Kästchen für Kästchen abfahren, um die Farbunterschiede zu registrieren. Dies ist sehr zeitaufwendig und würde ein kleines Spielfeld erfordern. Damit die Ausgangsposition, von welcher der Roboter beginnt seine Befehle auszuführen, ebenfalls konstant ist, wird eine Art Stopper an die Platte gebaut. Steht der Roboter mit der Hinterachse am Stopper, so zeichnet er das Feld garantiert im Aufnahmebereich der Kamera.

Um das Zeichnen auf der unebenen Oberfläche der LEGO-Platte zu ermöglichen, empfiehlt es sich eine Pappe im Zeichenbereich unterzulegen und an der Platte zu befestigen. Die Anordnung des Roboters, sowie diverser anderer beschriebener Elemente, ist skizzenhaft der Abbildung 2 zu entnehmen.

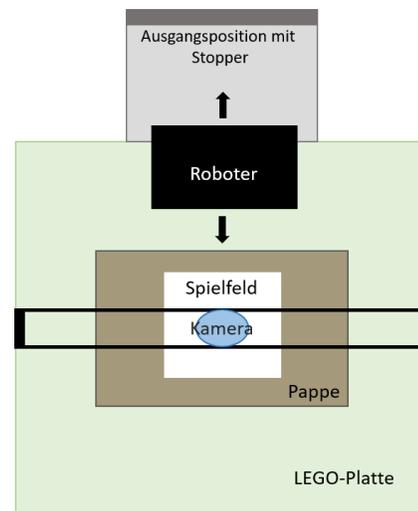


Abbildung 2: Skizze des Spielfeldes

B. Programmierung

Die Programmierung des Roboters erfolgt mit MATLAB. Der Spieler bedient den Roboter über ein Graphical User Interface kurz GUI. Die GUI enthält einen Startknopf, der das Programm startet und den Roboter aktiviert. Außerdem enthält die GUI neun Push-Buttons, die wie das Tic-Tac-Toe-Spielfeld angeordnet sind. So kann der Spielablauf überwacht werden. Es wird sichtbar, ob das vom menschlichen Spieler gesetzte Feld richtig erkannt wurde und ob der Computer das von ihm gewählte Feld auch tatsächlich richtig anfährt. Des Weiteren gibt es ein Ausgabefeld, in dem zum Ende des Spiels ausgegeben wird, ob der Spieler gewonnen oder verloren hat. Ein Beispiel, wie die GUI am Ende einer Spielrunde aussehen kann, ist die Abbildung 3.

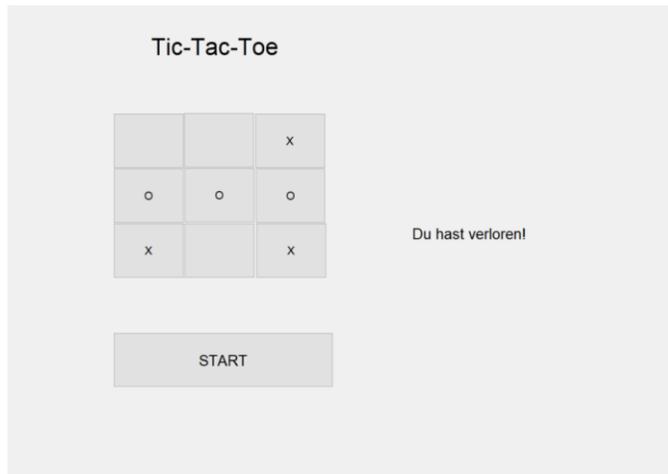


Abbildung 3: Graphical User Interface
 X – Vom Spieler gesetztes Feld
 O – Vom Roboter gesetztes Feld

Bevor das Programm gestartet wird, muss der Roboter in die Ausgangsposition gestellt werden, da die Anfahrtswege zum Spielfeld sowie zu den einzelnen Kästchen über den Befehl „Tacho Limit“ geregelt sind. Damit sind die Anfahrtswege immer identisch. Dabei muss beachtet werden, dass die Tacho-Limits der verschiedenen Motoren durch die Übersetzung des Motors für die Hinterachse nicht gleich sind. Horizontale und vertikale Linien sind einfach zu programmieren, da jeweils nur ein Motor angetrieben wird. Für die diagonalen Linien, die im Kreuz, welches der Roboter setzt, enthalten sind, müssen jedoch zwei Motoren gleichzeitig angesteuert werden. Auch hier muss beachtet werden, dass durch die Übersetzung verschiedene Werte genutzt werden müssen, um ein Kreuz zu erhalten. Nach dem Drücken des Startknopfes werden nacheinander die Motoren angesteuert, so dass der Roboter zuerst zum Spielfeld fährt und dieses dann zeichnet. Im Anschluss kehrt er in die Ausgangsposition zurück, welche sich außerhalb des Kamerabildes befindet. Sobald die Motoren gestoppt sind, wird das Spielfeld von der Kamera erfasst. Die einzelnen Kästchen des Spielfeldes sind fest im aufgenommenen Bild definiert. Daher ist die exakte Einhaltung der Kameraposition wichtig. Ansonsten würden die Felder verschoben oder nicht erkannt werden. Sobald das Bild aufgenommen ist, hat der Spieler zehn Sekunden Zeit seinen Zug zu machen. Danach wird ein zweites Bild aufgenommen. Die beiden Bilder werden in Grautöne umgewandelt und miteinander verglichen. Das Feld, in welchem die größte Veränderung vorliegt, wird als gesetzter Zug des menschlichen Spielers interpretiert. Damit dies zuverlässig funktioniert, dürfen sich die Lichtverhältnisse innerhalb der zehn Sekunden nicht verändern. Des Weiteren wird das gewählte Kästchen besser erkannt, je mehr des Kästchens ausgemalt ist. Es empfiehlt sich daher für den Spieler auf die typischen Zeichen (Kreuz bzw. Kreis) zu verzichten, um Fehlinterpretationen seitens des Programmes zu vermeiden. Im Gegensatz dazu kann der Roboter ohne Probleme das Kreuz beziehungsweise den Kreis verwenden, da seine Spielzüge nicht mit der Kamera erfasst werden, sondern vom Spielalgorithmus vorgegeben und abgespeichert werden.

Außerdem sollte darauf geachtet werden, dass der Spieler am

Ende seines Zuges sowohl den Stift als auch seine Hand aus dem Bereich des Kamerabildes nimmt. Das als vom Spieler gesetzt erkannte Feld wird in der GUI gekennzeichnet. Der Push-Button, welcher dem gesetzten Kästchen entspricht, wird mit einem „x“ versehen. Im Anschluss ermittelt der Spielalgorithmus, an welche Stelle der Roboter seinen Zug macht. Der entsprechende Push-Button wird dann auf „o“ gesetzt. Die Feldposition wird an den Roboter weitergegeben und dieser fährt das jeweilige Kästchen an. Jedes Kästchen des Feldes hat eine zugewiesene Nummer und auf diese Nummer jeweils einen eigenen programmierten Anfahrtsweg. Hat der Roboter sein Kästchen erreicht, so setzt er ein Kreuz in dieses. Bei der Programmierung wurde darauf geachtet, dass das Kreuz an der gleichen Ecke des Kästchens beginnt und endet, damit bei der folgenden Rückkehr in die Ausgangsposition keine zusätzlichen Verschiebungen entstehen.

Beachtet werden muss, dass in der GUI der Roboter den Kreis und der Spieler das Kreuz hat. Da jedoch das Programmieren des Ablaufes für ein Kreuz wesentlich unkomplizierter ist, setzt der Roboter auf dem tatsächlichen Spielfeld das Kreuz und der Spieler malt am besten das gesamte Kästchen aus. Nach der Rückkehr in die Ausgangsposition beginnt der Ablauf erneut ab dem Punkt, wo das erste Kamerabild aufgenommen wird. Abgebrochen wird dann, wenn die Schleife viermal wiederholt wurde. Das liegt daran, dass das Feld aus neun Kästchen besteht und somit maximal neunmal gesetzt werden kann. Bei zwei Spielern, die immer abwechselnd setzen, sind das fünf Züge für denjenigen der anfängt und vier Züge für den zweiten Spieler. Da immer der menschliche Spieler beginnt, kann der Roboter bei einem Unentschieden nicht mehr als viermal setzen und somit muss die Schleife viermal durchlaufen werden. Der gesamte Ablauf ist in der Abbildung 4 noch einmal schematisch dargestellt.

Für den Spielalgorithmus werden den gesetzten Kästchen Werte zugeordnet. Die Züge des Spielers erhalten den Wert „1“ und die Züge des Roboters den Wert „4“. Diese Werte werden in einer 3x3-Matrix abgespeichert. Die Matrix ist zum Beginn einer Spielrunde mit Nullen gefüllt. Nach jedem Spielzug, in dessen Folge sich die Werte ja ändern, werden jeweils die Spalten, Zeilen und Diagonalen auf ein Neues zusammengerechnet. Zuerst werden die erhaltenen Ergebnisse überprüft, ob sie den Wert acht annehmen. Ist dies der Fall, bedeutet das, dass der Roboter in einer Zeile, Spalte oder Diagonalen bereits zweimal gesetzt hat und mit seinem Zug nun gewinnen kann. Also wird die Nummer des leeren Kästchens an den Roboter weitergegeben. Ist dies nicht der Fall, wird als nächstes geprüft, ob einer der zusammengerechneten Werte zwei ist. Das würde bedeuten, dass der Spieler zwei gleiche Zeichen hat und gewinnen könnte. Dann würde ebenfalls die Position des dritten freien Kästchens weitergegeben werden, um den Sieg des menschlichen Spielers zu verhindern. Sollte beim Rechnen jedoch eine andere Zahl als zwei oder acht herauskommen, so setzt der Roboter in ein noch nicht belegtes Feld. Der Spielalgorithmus, das Zeichnen des Feldes, das Setzen des Kreuzes sowie das Aufnehmen und Verarbeiten des Kamerabildes bilden jeweils eigene Funktionen, die letztendlich zu einer Funktion zusammengefasst werden.

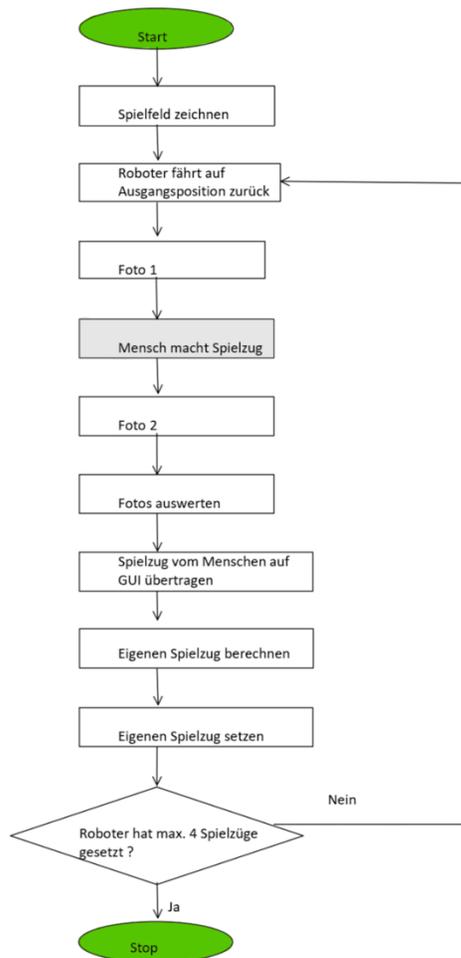


Abbildung 4: Programmablaufplan

C. Ergebnisse

Das Ergebnis des Projektes ist ein Roboter, der in der Lage ist, selbstständig mittels eines Stiftes ein Spielfeld zu zeichnen, die einzelnen Kästchen gezielt anzusteuern und sinnvolle Spielzüge zu machen. Diese können, wie auch der restliche Spielverlauf, mittels der erstellten GUI kontrolliert und nachvollzogen werden können. Die Kamera ermöglicht es dem Roboter mit dem Spieler zu interagieren, da sie dessen Züge erfasst.

Eine Spielrunde mit dem Ergebnis „unentschieden“ dauert etwa sechs Minuten. Gibt es bei einer Runde einen Gewinner, so werden insgesamt weniger Züge gemacht und somit verkürzt sich die Spielzeit.

IV. FAZIT UND AUSBLICKE

Das gewünschte Ziel des Projektes, einen funktionierenden Tic-Tac-Toe-Roboter zu bauen und zu programmieren, ist gelungen.

Die anfangs angestrebte Interaktion mit dem menschlichen Spieler funktioniert so, wie es geplant war. Sie kann aber noch verfeinert werden, indem sowohl Spieler als auch Roboter dem jeweils anderen signalisieren, wann ihr Zug beendet ist. Die festgelegte Reaktionszeit von zehn Sekunden des Spielers kann durch einen Taster ersetzt werden. Somit hat der Spieler die Möglichkeit genau zu überdenken, welchen Zug er machen möchte. Auch der Roboter kann mittels eines Signaltones nach dem ersten Kamerabild dem Spieler zeigen, dass er fertig ist und der Spieler am Zug ist. Diese Zusätze können das Spiel flüssiger machen und die Spielzeit somit verkürzen. Die Erfassung und Auswertung des Spielfeldes mittels Kamera erfolgt zuverlässig, solange genug Fläche des Kästchens ausgefüllt wird und die Kamera ihre Position exakt beibehält. Um das Problem mit der Kameraposition zu vermeiden, kann die Programmierung der Kamera so geändert werden, dass sie sich jedes Mal neu auf das jeweilige Spielfeld kalibriert.

Um die Spieloptionen zu erweitern, kann der Spielalgorithmus erweitert werden, so dass nicht nur der Spieler, sondern auch der Roboter das Spiel beginnen kann.

Zusammenfassend kann das Projekt als gelungenes Grundkonzept angesehen werden, welches mit kleinen Änderungen noch optimiert werden kann.

V. LITERATURVERZEICHNIS

- [1] J. K. Brickworks: LEGO Telegraph Machine and Printer. <https://jkbrickworks.com/telegraph-machine-and-printer>.
Version: März 2019
- [2] Michalis Koulouras: Beispiel für Sensorerkennung. <https://www.youtube.com/watch?v=x1Q8h7qegjk>.
Version: März 2019
- [3] Dave Corboy: Beispiel für Kameraerkennung. <https://www.youtube.com/watch?v=IfKaSQDMjYM>.
Version: März 2019