

Flunky-Ball-Roboter

Artem Kulyhin, Elektrotechnik und Informationstechnik
Otto-von-Guericke-Universität Magdeburg

Zusammenfassung—Im Rahmen des LEGO-Mindstorms-Workshop-Projekts wurde ein Flunky-Ball-Roboter gebaut. Der Roboter basiert auf NXT3 und besteht aus Motoren, einem Farbsensor und LEGO-Teilen. Der Code wurde in Matlab erstellt. Der Roboter kann von zwei oder mehr Personen gespielt werden, kann aber auch alleine gespielt werden.

Schlagwörter—LEGO Mindstorms, NXT, Projektseminar, Flunky-Ball-Roboter, Spielroboter.

I. EINLEITUNG

IN der heutigen Welt hat jeder das Recht, sich nach einem harten Arbeitstag zu entspannen und mit Freunden Spaß zu haben. Es gibt viele Spiele, die dazu dienen, Menschen zu unterhalten und Zeit miteinander zu verbringen. Bei solchen Spielen ist es nicht wichtig, ob man gewinnt oder verliert, da sie alle dazu da sind, Freundschaften zu schließen. Allerdings gibt es auch Spiele, bei denen Gewinnen oder Verlieren direkte Auswirkungen auf die Gesundheit haben kann. Es handelt sich um das Bier-Pong. Diejenigen, die schon einmal Zeit mit Freunden bei dieser Aktivität verbracht haben, werden in diesen Worten sicherlich einen Widerhall finden: Je öfter der Ball das Ziel trifft, desto gesunder ist der Werfer.

Der Flunky-Ball-Roboter wurde entwickelt, um die Genauigkeit von Tischtennisballwürfen zu verbessern. Der Ball und der zu treffende Becher entsprechen den Dimensionen des Spiels. Die Bewegungen des Roboters im Raum und die Drehung des Bechers sollen das Treffen erschweren.

II. VORBETRACHTUNGEN

A. Spielregeln

Um wettbewerbsfähig zu sein, wurden zunächst die Spielregeln entwickelt. Ein Farbsensor ermöglicht das Spielen in Teams. Das Spiel wird von einem blauen und einem roten Team mit jeweils farblich passenden Bällen gespielt. Die Spieler befinden sich auf gegenüberliegenden Seiten und bewegen sich jeweils einen Meter vom Roboter entfernt. Wenn der Ball den Becher trifft, erkennt der Sensor die Farbe und gibt ein Signal aus, das für jedes Team unterschiedlich ist. Danach endet das Spiel und man kann von vorne beginnen.

Der Roboter ist so programmiert, dass er sich innerhalb eines 0,8 m bis 0,8 m großen Quadrats auf einer vorgegebenen Bahn bewegt. Der Wettbewerb dauert 60 s. Wenn in dieser Zeit niemand den Ball wirft, beginnt das Spiel von vorne.

B. Konzeptuelle Feinheiten

Das gesamte Projekt wurde durch LEGO Mindstorms ermöglicht. Obwohl es eine Chance für die Phantasie bot,



Abbildung 1. Aussehen des Spielroboters

hatte es auch einige Nachteile. Die Zuverlässigkeit der Kunststoffmaterialien machte mehrere Umgestaltungen erforderlich. Außerdem gab es Einschränkungen in Form von drei Motoren und mehreren Sensoren (siehe Abb. 1). Der Mangel an Teilen erforderte die Zusammenarbeit mit anderen Teams, was die Kooperation vertiefte. Obwohl der Farbsensor nicht perfekt ist und die Farbe nicht immer erkennen konnte, wurde dennoch eine Lösung gefunden. Diese soll im Folgenden bestätigt werden.

C. Software

Der Code wurde in MATLAB geschrieben und war dank des von den Kursleitern zur Verfügung gestellten Materials verständlich. Wir konnten unsere im Informatikunterricht erlernten Programmierkenntnisse erfolgreich anwenden. Es war auch nützlich, die im Programmierunterricht erworbenen Kenntnisse anzuwenden. Der Computer wurde über das USB-B-Kabel an den Smartblock angeschlossen. Nach der Arbeit mit MATLAB stellte sich heraus, dass das Programm selbst nicht ideal war. Eine Schwierigkeit bestand darin, dass einige Elemente, insbesondere der Farbsensor, nicht konstant betrieben werden konnten. Die Lösung bestand darin, den Sensor von Bewegung zu Bewegung zu aktivieren, anstatt ihn kontinuierlich zu betreiben.



Abbildung 2. Erste Ansicht des raupenbasierten Roboters

III. PROJEKTREALISIERUNG

A. Anfängliches Projekt und Verbesserung

Zu Beginn wurde beschlossen, den Roboter auf einer Raupenkettengrundbasis aufzubauen, um ihn stabiler zu machen (siehe Abb. 2). Allerdings hatte dies den gegenteiligen Effekt: Die Raupen rollten nach innen und die Krümmung des Roboters war sogar mit bloßem Auge erkennbar. Zudem wurde ein grundlegender Fehler festgestellt. Aufgrund der Besonderheiten bei der Arbeit mit Lego ist nicht sofort klar, wie der intelligente Stein befestigt werden soll. Die Schwierigkeit besteht darin, dass er schwer ist und sich durch die beweglichen Teile hindurchdrückt.

Nach Überlegungen wurden zwei wichtige Entscheidungen getroffen: Erstens sollte alles um den intelligenten Stein herum gebaut werden, da er der monolithischste und stärkste Teil ist. Zweitens ist es notwendig, auf eine Radbasis umzusteigen. Obwohl dadurch der Kontakt mit der Oberfläche verringert wird, erhält der Roboter aufgrund der fehlenden Drehmomentübertragung Geschwindigkeit und Mobilität, was für das Projekt notwendig ist.

B. Technisches Teil

Das gesamte Projekt ist mobil dank drei Motoren, für die jeweils ein eigener Code geschrieben wurde. Die Motoren A und B wurden für die Bewegung verwendet. Jeder Motor war nur für ein Rad zuständig, so dass der Roboter sich auf der Stelle drehen und wenden konnte. Zur Stabilisierung wurde ein Element verwendet, das einem Lager ähnelte. Es war am vorderen Teil des Roboters angebracht und ermöglichte auch Drehbewegungen, da es sich in seiner Hülle in alle Richtungen drehen konnte. Der Roboter blieb auch bei maximaler Leistung und einem scharfen Start stabil und behielt seine Position bei. Um eine bessere Haftung auf der Oberfläche zu gewährleisten, wurden zwei Räder auf jeder Seite verwendet. Es wird jedoch empfohlen, den Roboter nur auf einer ebenen und harten Oberfläche zu verwenden.

Motor C ist für die Drehung des Bechers verantwortlich. Aufgrund der geringen Größe des Roboters wurde beschlossen, die Drehung bei 195° zu belassen. Die Bewegung des Bechers

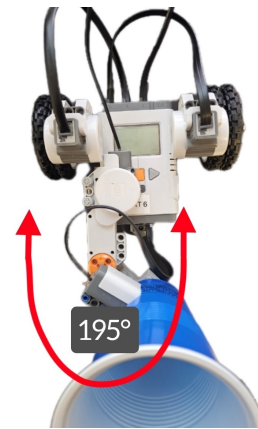


Abbildung 3. Die Mobilität des Roboters

beginnt an der äußersten rechten Position und setzt sich fort durch eine Drehung um 195° gegen den Uhrzeigersinn, gefolgt von einer Drehung um 195° im Uhrzeigersinn und so weiter während des gesamten Spiels. (siehe Abb. 3) Durch die Bewegung des Bechers wird es viel schwieriger, das Ziel zu treffen. Tests haben gezeigt, dass die Genauigkeit des Balls um 25 % abnimmt, wenn der Becher bewegt wird. Es gibt 4 Verbindungskabel, die die Motoren und den Sensor mit dem Gerät verbinden.

C. Farbsensor-Anwendung

Ein Farbsensor ist am Becher befestigt. Wenn der Ball auf den Becher trifft, erkennt der Sensor die Farbe des Balls genau, da der Abstand zwischen Sensor und Ball nur wenige Millimeter beträgt. Ein roter Ball wird mit 800 Hz signalisiert, ein blauer mit 300 Hz. So kann man erkennen, welche Mannschaft den Ball geworfen hat und somit das Spiel gewonnen hat. Die Tondauer beträgt 760 Millisekunden, was ausreichend ist, um den Sieger deutlich zu hören, ohne dass der Ton störend wirkt.

Das Hauptproblem besteht darin, dass der vorhandene Sensor nicht in der Lage ist, den Inhalt des Bechers kontinuierlich abzutasten. Er kann nur ein- und ausgeschaltet werden, aber nicht die ganze Zeit eingeschaltet bleiben. Die Lösung des Problems war nicht der ständige Betrieb des Sensors, sondern das ständige Ein- und Ausschalten des Sensors. Der Sensor wird aktiviert, wenn der Roboter eine Bewegung ausführt, wie z.B. Drehen, Anhalten oder Rotieren.

```
OpenNXT2Color(port , 'FULL')
color = GetNXT2Color(port)
if strcmp(color , 'BLUE')
    NXT_PlayTone(300 , 760)
    StopMotor('all' , 'off')
    break
elseif strcmp(color , 'RED')
    NXT_PlayTone(800 , 760)
    StopMotor('all' , 'off')
    break
end
```

Diese Lösung hat einen Nachteil: Der Sensor funktioniert nur, wenn sich die Bewegung ändert, nicht während der Bewegung selbst. Wenn der Ball also während der Vorwärtsbewegung geworfen wird, erreicht der Roboter erst den Wendepunkt, bevor das Spiel endet. Die beste Änderung am Matlab-Programm wäre die Möglichkeit, mehrere verschiedene Codes parallel zu schreiben und auszuführen. Das bedeutet, dass ein Code für die Bewegung des Roboters und ein anderer für die Steuerung der Sensoren zuständig ist. Dies würde es ermöglichen, den bestehenden Code zu vereinfachen und unsere Idee mit dem kontinuierlichen Abtasten des Becherinhalts zu realisieren.

D. Programm

Der Gesamtcode besteht aus Zyklen. Zunächst beschreibt der Roboter ein Quadrat und führt dann innerhalb dieses Quadrats weitere Aktionen aus. Anschließend werden alle Aktionen im Hauptzyklus wiederholt.

Wie bereits erwähnt, dreht sich der Becher kontinuierlich. Dies wird durch eine von NXT gelesene Zeile ermöglicht. Sie prüft, ob der Motor C läuft und schaltet ihn gegebenenfalls wieder ein, damit die Drehung in die entgegengesetzte Richtung geht und der Becher stets in Bewegung bleibt.

```
OBJC = NXTMotor( 'C', 'Power', 10)
x=0
while x < 2
    i=0
    while i < 4
        if ~OBJC.ReadFromNXT().IsRunning
            OBJC.Power = -OBJC.Power;
            OBJC.TachoLimit = 180;
            OBJC.SendToNXT()
            OBJA = NXTMotor( 'A', 'Power', 50)
            OBJB = NXTMotor( 'B', 'Power', 50)
            OBJA.TachoLimit = 600;
            OBJB.TachoLimit = 600;
            OBJA.SendToNXT()
            OBJB.SendToNXT()
            OBJA.WaitFor()
            OBJB.WaitFor()
```

E. Bluetooth-Anwendungspotenzial

Viele Betriebsprobleme sind auf die Länge des Kabels zurückzuführen. Der Spielroboter ist sehr mobil und die Länge des Kabels schränkt sein Potenzial stark ein. Nachdem es möglich war, den Roboter über Bluetooth zu verbinden, zeigten Tests überraschende Ergebnisse. Das Spiel war interessant und der Roboter konnte sich frei bewegen. Jedoch hat der NXT den Nachteil, dass die Bluetooth-Verbindung nach drei Versuchen vollständig verloren geht und nur noch ein kurzes Kabel als Ausweg bleibt. Trotzdem wurde bei den kurzen 3 Versuchen über Bluetooth festgestellt, dass es eine Verzögerung zwischen dem Senden der vom Sensor gelesenen Daten an den Computer und dem Stoppen des Spiels gibt. Diese Verzögerung entspricht der Verzögerung vor dem Start des Projekts über Bluetooth. Es handelt sich um einen Fehler von MATLAB, der leider nur durch eine Änderung des Programms selbst behoben werden

kann. Das Potenzial der Verwendung von Bluetooth in dem Projekt ist jedoch enorm.

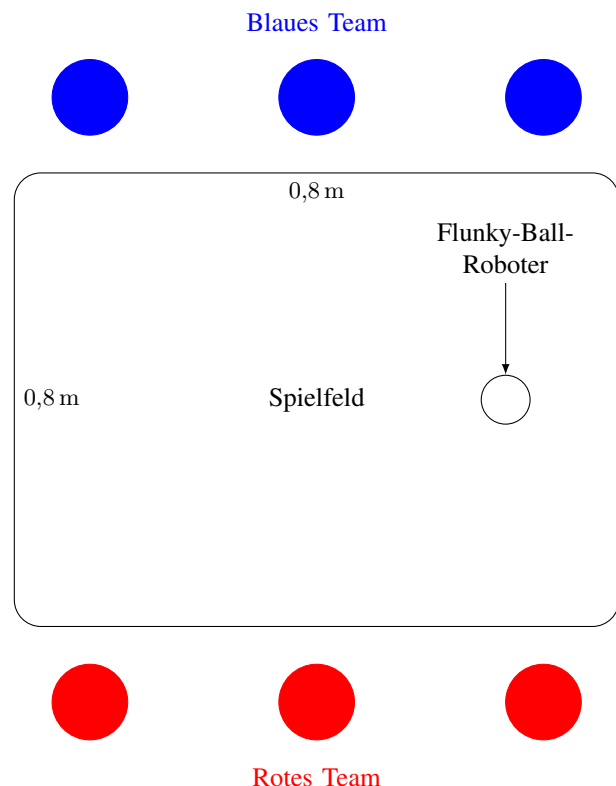


Abbildung 4. Das Spielschema "Precision Shooter".

F. Precision Shooter

Precision Shooter ist der Name des ersten Spiels, das für diesen Roboter entwickelt wurde. Es ist ein lustiges Spiel für einen angenehmen Zeitvertreib in Gesellschaft. Für das Spiel benötigt man zwei Personen und die gleiche Anzahl Bälle in zwei Farben (siehe Abb. 4). Der Roboter steht in der Mitte des Raumes und die Spieler bewegen sich 2 Meter von ihm entfernt. Nachdem der Roboter gestartet wurde, haben die Spieler 1 Minute Zeit, um den Ball in den Korb zu werfen. Die Mannschaft, die den Ball zuerst in den Korb wirft, gewinnt. Schafft es keine Mannschaft, den Ball in der vorgegebenen Zeit in den Korb zu werfen, beginnt das Spiel von vorne. Es ist auch möglich, die Spielmechanik des Roboters auf verschiedene Weise zu verändern. Zum Beispiel kann man die Anzahl der Teams auf 3, 4 oder mehr erhöhen, indem man dem Code nur 2 Zeilen hinzufügt.

Natürlich ist es auch möglich, Precision Shooter alleine zu spielen. Dazu stellt der Spieler den Roboter in die Mitte des Raumes, entfernt sich 2 Meter und versucht dann, den Ball in den Becher zu werfen. Dieses Spiel ist nicht nur für Erwachsene, sondern vor allem auch für Kinder geeignet, da es die Koordination und Zielgenauigkeit schult.

G. Flunky-Ball – neue Interpretation

Der Roboter ist auch ideal, um Kindern und Erwachsenen das alte Spiel Flunky Ball beizubringen. Zwei Mannschaften

stehen sich in einem Abstand von ca. 3-4 Metern gegenüber. Die Anzahl der Spieler pro Team ist nicht standardisiert, in der Regel haben die Teams zwischen 3 und 8 Spieler. Vor jedem Spieler auf dem Boden steht eine Flasche mit Wasser, Limonade oder einer anderen Flüssigkeit. Zwischen den beiden Mannschaften steht ein Roboter in der Mitte des Spielfeldes. Beide Mannschaften haben gleich viele Bälle und werfen sie abwechselnd. Trifft eine Mannschaft, muss ein Spieler seine Flasche austrinken, dann den Ball aus dem Becher holen und das Spiel geht weiter. Nach einem Treffer trinkt der nächste Spieler. Wenn eine Mannschaft alle Getränke ausgetrunken hat, hat sie das Spiel gewonnen.

IV. ERGEBNISDISKUSSION

Das Endergebnis des Projekts wurde erfolgreich getestet und bei der Abschlusspräsentation vorgestellt. Das vollständige Video des Spiels ist auf dem YouTube-Kanal von Kursleiter Mathias Magdowski zu finden [1]. Das Spiel dauert eine Minute und bietet die Möglichkeit, es mehrmals zu spielen, und der Roboter entwickelt Wurf-Fähigkeiten. Die Möglichkeiten, den Code und das Projekt in Zukunft zu nutzen, sind vielfältig. Leider gibt es in MATLAB eine Reihe von Problemen, die es nicht ermöglichen, das Potenzial des Roboters voll auszu-schöpfen. Um viele derzeit ungenutzte Ideen zu verwirklichen, können die folgenden Änderungen in MATLAB vorgenommen werden:

- 1) Paralleles Schreiben von Code für mehrere Elemente gleichzeitig. Am Beispiel des Farbsensors wird die Notwendigkeit des kontinuierlichen Betriebs bestimmter Mechanismen.
- 2) Verbesserung der Bluetooth-Verbindung. Die aktuelle Bluetooth-Verbindung ist instabil und funktioniert nicht richtig. Eine Lösung dieses Problems kann den Anwendungsbereich der Technologie, einschließlich des entwickelten Roboters, erheblich erweitern.

V. ZUSAMMENFASSUNG UND FAZIT

Insgesamt haben die Leiter und Teilnehmer des Kurses in den zwei Wochen ein enormes Arbeitspensum bewältigt. Alle Probleme wurden gelöst und das Endergebnis ist erfreulich und inspirierend. Die Otto-von-Guericke-Universität sollte solche Kurse häufiger anbieten, da sie die Fähigkeiten der Teilnehmer tatsächlich fördern und ihnen die Möglichkeit zur Selbstverwirklichung bieten.

LITERATURVERZEICHNIS

- [1] Mathias Magdowski: *YouTube*.
<https://youtu.be/CqgnKNTeBPU?t=17910> Version: Februar 2024
- [2] WIKIPEDIA: *Flunkyball*
<https://de.wikipedia.org/wiki/Flunkyball> Version: Februar 2024