

Regelung eines selbstbalancierenden Roboters

Saeid Miri, MTK

Otto-von-Guericke-Universität Magdeburg

Zusammenfassung—Der zweirädrige Balanceroboter ist zu einem sehr beliebten Projekt in den Bereichen Mechatronik und Robotik geworden. Durch die Nutzung des umgekehrten Pendels gehört es zu den klassischen Systemen in Dynamik und Regelung. Allerdings stellt dieses System aufgrund seiner inhärenten Instabilität und Nichtlinearität eine der herausforderndsten Probleme in der Regelungstechnik dar. Die Stabilität des Roboters wird durch eine Kombination aus mechanischem Design, Rädern, Motoren, Sensoren und fortschrittlicher Software erreicht, die es ihm ermöglicht, sein Schwerpunkt in Echtzeit anzupassen. Das Projekt ist in zwei Hauptgruppen unterteilt: Hardware, Softwareimplementierung. Es wird eine detaillierte Betrachtung des Fortschritts jeder Phase erfolgen.

Schlagwörter—Gyroskop-Sensor, zweirädriger Roboter, PID-Regler, Regelungstechnik, Self-balancing Robot, Segwayroboter

I. EINLEITUNG

In den letzten Jahren hat die Robotik weltweit bedeutende Fortschritte gemacht. Die Durchführung von internationalen Robotikwettbewerben in verschiedenen Bereichen sowie die Veröffentlichung wertvoller Arbeiten zur Robotik sind ein Beweis für diese Behauptung. Das Konzept des Balance-Roboters basiert auf dem umgekehrten Pendel [1]. Im Gegensatz zu einem gewöhnlichen Pendel neigt ein umgekehrtes Pendel dazu, instabil zu sein.

Während des Lego-Mindstorms-Projekts sollte ein zweirädriger selbstbalancierender Roboter gebaut werden, der sich mithilfe von Sensoren und Aktoren aufrecht hält und stabil auf seiner eigenen Achse bewegen werden kann. Im Vergleich zu Robotern mit drei oder vier Rädern benötigen sie weniger Komponenten und belegen weniger Platz. Aus diesem Grund haben zweirädrige Roboter einen größeren Freiheitsgrad und sind in kleinen Räumen äußerst nützlich. Eine gute Idee könnte die Verwendung dieser Roboter als Rollstuhl sein [2]. Außerdem sollte dieser Roboter in der Lage sein, einer Linie zu folgen und auf ihr hin- und zurückzufahren. Die Linienverfolgung konnte aufgrund von Zeitbeschränkungen leider nicht implementiert werden.

Da im Bereich der Mechatronik viel mit Steuerung und Regelung gearbeitet wird, ist dieses Projekt besonders relevant und lehrreich für uns.

II. VORBETRACHTUNGEN

Roboter mit zwei Rädern weisen aufgrund ihrer Physik einige Vorteile auf. Auf Basis von ihrem geringeren Gewicht und ihrer wenigeren Komponenten haben diese Roboter einen niedrigeren Energieverbrauch und können zu einer guten Option im Transportsystem werden. Segway ist ein praktisches

und kommerzielles Beispiel für diesen Roboter, welche die Neigungswinkel durch den Sensor ständig kontrolliert würde.

Es wurde von uns ein Roboter gebaut, der auf denselben Prinzipien basiert und durch den Gyro-Sensor wird den Winkel ständig gemessen. Um das Verständnis des selbstbalancierenden Roboters zu verbessern, ist es erforderlich, zunächst das Konzept des Selbstausgleichsmodus, des Gyrosensors und des Regelkreises zu verstehen.

A. Selbstausgleichsmodus

Der Selbstausgleichsmodus ist das beeindruckendste Merkmal von zweirädriger Robotern. Um zu verstehen, wie dieses System funktioniert, kann man feststellen, dass es eine ähnliche Funktion wie der menschliche Körper aufweist. Das Gleichgewicht geht verloren, wenn aus einer aufrechten Position nach vorne gelehnt wird, aber in der Regel wird man dabei nicht flach auf das Gesicht fallen. Nachdem das Gleichgewicht verloren wurde, beginnt das Gehirn wahrzunehmen und das Bein nach vorne zu bringen, um ein Fallen zu verhindern. Ein zweirädriger Roboter macht fast das gleiche, außer dass er anstelle von Beinen Räder hat, anstelle von Muskeln einen Motor und anstelle von Gehirn eine Sammlung von Mikroprozessoren und Sensoren hat. In Abbildung 1 ist zu erkennen, dass der Roboter aufgrund des Drehmoments nach vorne zu kippen droht. Dieses Drehmoment entsteht durch die Gewichtskraft $mg \cdot \cos(\theta)$. Um dem entgegenzuwirken und den Roboter aufrecht zu halten, werden die Motoren verwendet.

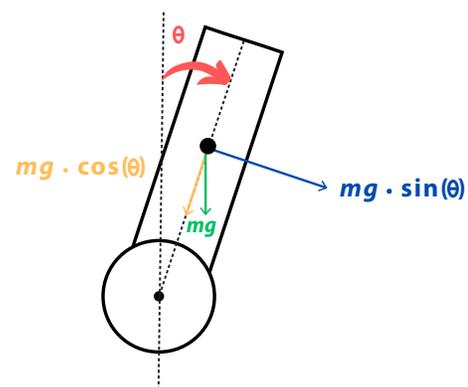


Abbildung 1. Self-balancing Robot model

B. Gyrosensor

Ein Gyroskop erfasst die Rotationsgeschwindigkeit und den Neigungswinkel um eine Achse. Die Drehung um eine Achse wird normalerweise in Grad pro Sekunde gemessen und eine Rotation in entgegengesetzter Richtung erzeugt ein Signal

mit umgekehrtem Vorzeichen. Wenn die Rotation gestoppt wird, wird ein Wert von Null gemessen. Gyrosensoren können neben der Erfassung der Rotationsgeschwindigkeit auch die Bewegung des Körpers messen. Um Bewegungen genau zu messen, werden Gyrosensoren oft mit Beschleunigungssensoren kombiniert.

C. Regelkreis

Ein Regelkreis hat den Zweck, automatisch ein stabiles System zu generieren und sicherzustellen, dass das gewünschte Ausgangssignal oder die Referenz erreicht wird. Das System misst kontinuierlich den Ist-Zustand eines Prozesses und vergleicht ihn mit einem Soll-Zustand. Falls eine Abweichung erkannt wird, wird das System mittels einer Rückkopplung angepasst, um die Abweichung zu minimieren und den Prozess in Richtung des Soll-Zustands zu steuern.

III. HARDWARE

A. Aufbau

Der selbstbalancierende Roboter basiert hauptsächlich auf einem LEGO-Stein, welche über einen leistungsstarken ARM9-Prozessor verfügt [3], der schnelle Berechnungen und komplexe Programme ermöglicht. Zusätzlich verfügt er über einen Sensor, um die Position des Roboters zu messen. Zudem sind zwei Elektromotoren verbaut, die mit einer Geschwindigkeit von 160 U/min bis 170 U/min und einem Drehmoment von 20 N cm laufen [3]. Beim Bau des Roboters müssen viele Faktoren berücksichtigt werden, um eine stabile und gut steuerbare Konstruktion zu gewährleisten. Eine wichtige Überlegung ist eine gleichmäßige Gewichtsverteilung und der Schwerpunkt muss in der Mitte und nahe am Boden liegen. Im ersten Prototyp wurde der EV3-Stein vertikal und weit oben positioniert, was zu einer instabilen Konstruktion und erhöhte das Risiko des Umkippens. Bei dem zweiten Prototyp wurde der Stein im unteren Bereich des Roboters angebracht, aber aufgrund des vertikalen Aufbaus des Steins war das Gewicht ungleichmäßig verteilt. Letztendlich wurde, wie in Abbildung 2 zu sehen ist, der EV3-Stein horizontal auf der Radachse positioniert, um den Schwerpunkt nah am Boden zu halten und das Gewicht gleichmäßig zu verteilen. Zusätzlich sollte bei der Auswahl der Räder darauf geachtet werden, dass größere Räder verwendet werden, um aus dem Hebelgesetz (1) [4] ein höheres Drehmoment zu erzeugen. Diese Konstruktion sorgt für eine höhere Stabilität und eine bessere Steuerung des Roboters.

$$M = F \cdot r \quad (1)$$

B. Auswahl des Sensors

Für die Bestimmung der Position des Roboters wurde ursprünglich ein Lichtsensor verwendet, der aufgrund seiner Empfindlichkeit gegenüber Veränderungen in der Umgebungshelligkeit sowie Schattenwürfen durch den Roboter ungenauere Werte lieferte. Daher wurde anstelle des Lichtsensors ein Gyrosensor auf die Radachse gebaut, um den Winkel und die Drehgeschwindigkeiten des Roboters präzise zu messen und zu steuern. Obwohl der Gyrosensor im Laufe der Zeit aufgrund des Drifts ungenauere Werte liefert, erwies er sich als besser geeignet als der Lichtsensor.



Abbildung 2. Robot Aussehen

IV. SOFTWARE

A. Programmierung und Funktionsweise eines PID-Reglers

Ein PID¹-Regler ist ein kombinierter Regler, Der drei Komponenten basiert. Um den PID-Regler für einen selbstbalancierenden Roboter zu programmieren, müssen zunächst die richtigen Parameter für die drei Komponenten gefunden werden. Die integrale Komponente berücksichtigt die zeitliche Komponente des Fehlers, die derivativen Komponente berücksichtigt die Geschwindigkeit des Fehlers und die proportionale Komponente gibt die Größe der Reaktion des Reglers auf den Fehler an. Die Struktur des PID-Reglers, welche in Abbildung 3 dargestellt ist, zeigt, dass die Integral, Proportional und Differential-Steuerungen parallel arbeiten und am Ende ihre Signale addiert werden, um das Steuersignal zu erzeugen. Es ist selten erforderlich, eine integrale Größe zu verwenden. Diese Größe kann in der Regel vernachlässigt werden. Nur wenn die Beschleunigung ein wichtiger Faktor für den Roboter ist, wird diese Größe benötigt. Aus diesem Grund wird für den Balance-Roboter ein PD-Regler gegenüber einem PID-Regler bevorzugt. Um den PD-Regler auf einem EV3-Stein zu implementieren, wird es folgendermaßen vorgegangen:

$X \rightarrow$ Proportionalfehler

$Z \rightarrow$ Derivativfehler

¹Proportional, Integral, derivative

$$X = \text{gewünschter Winkel} - \text{aktueller Winkel} \quad (2)$$

$$Z = \text{ProportionalFehler} - \text{vorherige ProportionalFehler} \quad (3)$$

$$\text{Motor Leistung} = (K_P \cdot X) + (K_D \cdot Z) \quad (4)$$

Um das Verständnis der Programmierung zu verbessern, würde ein Programmablaufplan erstellt. Wie in Abbildung 4 deutlich zu sehen ist, wird der Winkel des Roboters auf 0 Grad gesetzt, um sicherzustellen, dass er in der Aufrechten Position bleibt. Zunächst wird der Winkel des Roboters durch den Sensor gemessen. Anhand der gemessenen Sensorwerte werden im nächsten Schritt die PD-Werte berechnet. Der Proportionalanteil des PD-Reglers wird durch den Unterschied zwischen dem gewünschten Winkel und dem aktuellen Winkel bestimmt, während der Differentialanteil auf der Rate des Winkeländerung basiert. Die Kombination der beiden Anteile ergibt die Steuergröße, die zur Ansteuerung des Motors verwendet wird, um den Roboter vorwärts und rückwärts zu bewegen. Dieser Ablauf wird kontinuierlich wiederholt, um den Roboter stabil und balanciert zu halten.

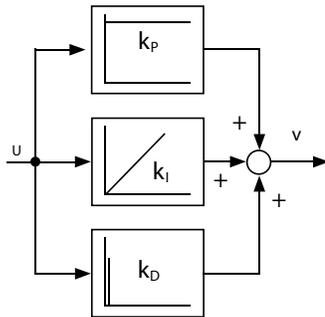


Abbildung 3. Blockdiagramm einen PID-Regler [5]

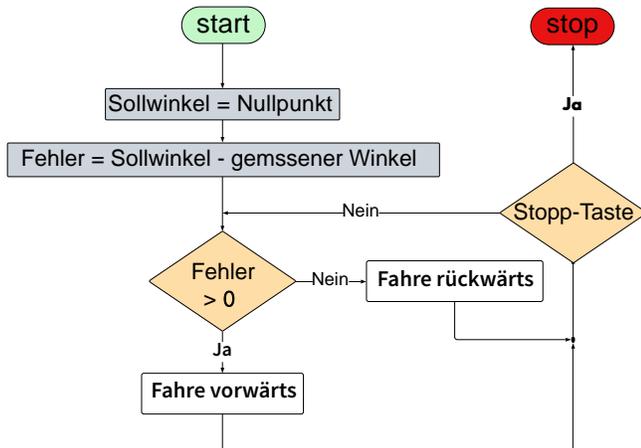


Abbildung 4. Programmablaufplan

B. Programmauswahl

Zu Anfang wurde MATLAB als Programmiersprache eingesetzt. Allerdings kam es aufgrund eines Kabels, das zwischen dem EV3-Stein und dem PC verbunden war, zu einer Störung im System, da das Kabel ständig den Roboter zog. Aufgrund dieser Störung war es schwierig, die exakten Parameterwerte für den PID-Regler auszuwählen. Nach Recherchen stellte sich heraus, dass das MATLAB-Skript nicht auf den Lego-Stein hochgeladen werden konnte. Darüber hinaus wurde als Ausgangswert des MATLAB-Skripts eine Geschwindigkeit erzeugt, obwohl für unsere Zwecke die Motorleistung benötigt wurde. Aus diesem Grund war die LEGO-Mindstorms Software die einzige Lösung, um das Programm reibungslos ausführen zu können, da die Daten direkt auf den LEGO-Stein hochgeladen werden konnten.

C. Parameterauswahl

Die Programmierung des PD-Reglers stellte eine Herausforderung dar, da es schwierig war, die optimalen Parameter für die zwei Komponenten zu bestimmen. Um den gewünschten Stabilitätsgrad zu erreichen, wurden mehrere Tests durchgeführt und die Parameter kontinuierlich angepasst. Wie aus Abbildung 5 ersichtlich wird, erhöhte sich im Laufe der Zeit die Fehleramplitude für die ausgewählten Werte und führte zu Schwingungen. Nach einigen Anpassungen konnten schließlich die korrekten Parameter gefunden werden, wodurch es möglich wurde, den Roboter erfolgreich selbstbalancierend zu machen, wie in Abbildung 6 dargestellt. Bei jeder Motorreaktion wurde die Error auf Null reduziert.

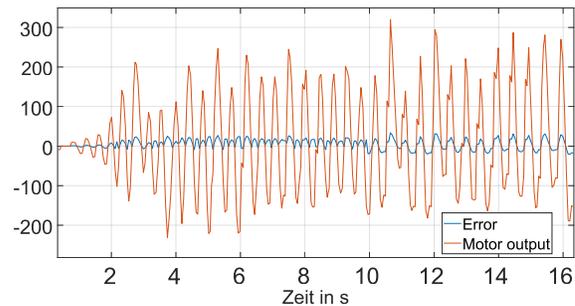


Abbildung 5. Graphische Darstellung des Fehlers und Motorleistung mit $K_P = 9$ und $K_D = 0,1$

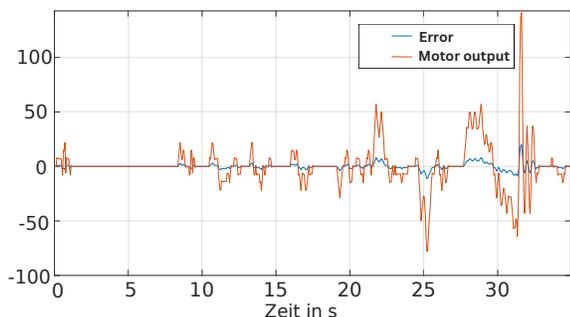


Abbildung 6. Graphische Darstellung des Fehlers und Motorleistung mit $K_P = 7$ und $K_D = 0,2$

V. ERGEBNISDISKUSSION

Letztendlich wurde ein selbstbalancierender Roboter gebaut, der in der Lage war, sich aufrecht zu halten. Es war jedoch erforderlich, vor jedem Start zu überprüfen, ob der gewünschte Winkel auf Null lag, da der Lego-Gyrosensor zu Drift neigte.

VI. ZUSAMMENFASSUNG UND FAZIT

Angesichts des Fortschritts der Technologie und des Einsatzes moderner Fahrzeuge ist es offensichtlich, dass die Welt in Richtung Fahrzeuge mit höheren Fähigkeiten und geringerem Verbrauch geht. Aus diesem Grund wurde eines der am häufigsten verwendeten und kostengünstigsten Fahrzeuge in kleinen Größen in dieser Studie untersucht, nämlich der Segway.

In diesem Artikel wurde Gyrossensor verwendet, um den Neigungswinkel des Roboters zu bestimmen. Als Regelsystem wurde zusätzlich ein PD-Regler eingesetzt, welcher in der Industrie weit verbreitet ist. Der PD-Regler gilt als bewährtes und zuverlässiges System zur Regelung und zeichnet sich durch seine einfache Struktur und hohe Effektivität aus. Eine Möglichkeit zur Entwicklung eines intelligenteren Balancing-Roboters in der Zukunft wäre die Kombination von Beschleunigungssensor und Gyrosensor zur Messung von Winkeln mit höherer Präzision. Eine andere Option wäre die Integration eines zweiten PID-Reglers, um einen zweirädrigen Selbstbalancing-Roboter in der Lage zu machen, sich möglicherweise in alle vier Richtungen (vorwärts, rückwärts, links und rechts) bewegen könnte. Des Weiteren könnte die Verwendung eines Lichtsensors in Kombination mit einem PID-Regler sein, eine Linienverfolgung zu ermöglichen.

LITERATURVERZEICHNIS

- [1] CHAN, Ronald Ping M. ; STOL, Karl A. ; HALKYARD, C R.: Review of modelling and control of two-wheeled robots. In: *Annual reviews in control* 37 (2013), Nr. 1, S. 89–103
- [2] GOHER, KM ; TOKHI, MO ; SIDDIQUE, NH: Dynamic modeling and control of a two wheeled robotic vehicle with a virtual payload. In: *ARPN Journal of Engineering and Applied Sciences* 6 (2011), Nr. 3, S. 7–41
- [3] TEAM, Lego: *Lego Mindstorm*, <https://education.lego.com/en-us/lessons/ev3-space/turn-using-sensor>
- [4] WIKIPEDIA: *Hebelgesetz*, [https://de.wikipedia.org/wiki/Hebel_\(Physik\)](https://de.wikipedia.org/wiki/Hebel_(Physik))
- [5] HEINRICH, Berthold ; LINKE, Petra ; GLÖCKLER, Michael ; HEINRICH, Berthold ; LINKE, Petra ; GLÖCKLER, Michael: Regelungstechnik. In: *Grundlagen Automatisierung: Erfassen-Steuern-Regeln* (2020), S. 123–229