

Konstruktion und Entwicklung eines Ball-Balance-Roboters

Prof. Dr.-Ing. Dietmar Jennewein, Dipl.-Ing. Alexander Russ, Matthias Nix, Simeon Rongstock

Ein Roboter, der sich selbst ohne Eingriff von außerhalb auf einem Ball ausbalancieren kann, war vor einigen Jahren noch Wunschdenken. Auch heute versetzt dieser Anblick sicherlich viele Leute in Staunen. Dank vorangeschrittener Technik in den Bereichen Sensorik und Elektronik lässt sich solch ein Vorhaben nun realisieren. Im Rahmen eines Forschungs-Projekts (IFP) innerhalb des Masterstudiengangs Maschinenbau wurde das Ziel gesetzt, solch einen Balance-Roboter zu bauen und ihn mit Hilfe eines Programmablaufs zum Balancieren zu bringen. Der Roboter setzt sich aus den Komponenten Schrittmotoren, Räder, Mikrokontroller, Beschleunigungssensor und der Grundkonstruktion zusammen.

Motivation

Um regelungstechnische Abläufe und Programme von der Theorie in die Praxis umsetzen zu können und eine Grundlage für weitere regelungstechnische Forschungsprojekte zu schaffen, wird dieses Projekt gestartet. Außerdem soll damit Schülerinnen und Schülern und technisch interessierten Personen auf Fachmessen ein Einblick in die Regelungs- und Elektrotechnik gegeben werden. Des Weiteren soll das Projekt zukünftig der Implementierung weiterer neuer Regel- und Sensorsysteme dienen.

Bauteilenauswahl und Konstruktion

Die Bauteile zur Realisierung des Roboters sind nach intensiver Recherche und Festlegen wichtiger Eckdaten ausgewählt worden.

Zu Beginn ist es erforderlich, einen passenden Motor zu finden, der bei geringer Baugröße ein optimales Drehzahl-Drehmoment-Verhältnis liefert und präzise anzusteuern ist. Dies ist für den Balanciervorgang besonders wichtig. Aufgrund dieser Randbedingungen werden Schrittmotoren eingesetzt, die über die Anzahl der Schritte oder das Vorgeben einer Drehzahl angesteuert werden können. Die Motoren besitzen eine Schrittweite von $1,8^\circ$ (200 Schritte pro Umdrehung) und ein Haltemoment von $1,10\text{Nm}$.

Damit der Roboter auf einem Ball balancieren und fahren kann, werden spezielle Räder benötigt, die rotatorisch angetrieben werden und eine translatorische Bewegung zulassen. Aus diesem Grund kommen sogenannte Omnidirectional-Wheels (Omni-Wheels) zum Einsatz, auf deren Umfang sich zusätzliche Rollen befinden. Die Rollen ermöglichen es, die zuvor genannten Anforderungen zu erfüllen. Die doppelte Ausführung der Räder dient dem gleichmäßigeren Abrollen, da durch diese Anordnung die Lücke des ersten Rades durch eine Rolle auf dem zweiten Rad überbrückt wird (siehe Abb. 1).

Um den Balanciervorgang des Roboters zu realisieren, werden programmierbare Mikrokontroller von Arduino eingesetzt. Zur Berechnung der erforderlichen Drehzahlen an den einzelnen Motoren und zur Datenübertragung wird der Arduino-Mega-2560 verwendet. Der Arduino-Uno und das Adafruit-Motor-Shield werden zur Ansteuerung der Motoren eingesetzt. Außerdem wird das Arduino-9-Axes-Motion-Shield zur Lagebestimmung des Roboters verwendet.

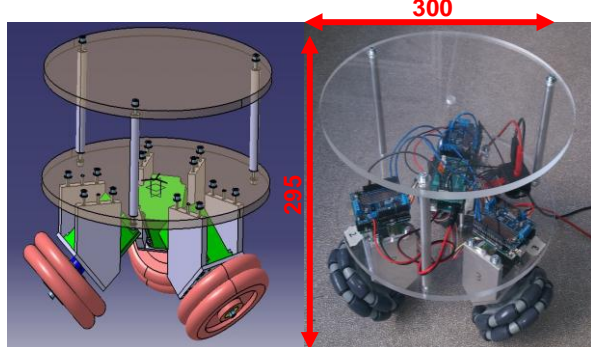


Abbildung 1: Vom CAD-Modell zum fertigen Roboter

Die Konstruktion des Roboters ist auf die Ab- und Montage Maße der ausgewählten Bauteile abgestimmt. Als Grundplatte und Befestigungsmöglichkeit der Bauteile des Roboters dient eine Plexiglasplatte mit einem Durchmesser von 300mm.

Die Motorhalterungen sind an das Bohrbild der Schrittmotoren angepasst. Außerdem sind die Winkel der Motorhalterungen so ausgelegt, dass die

verwendeten Omni-Wheels mit beiden Rädern an der Mantelfläche des Balls aufliegen.

Um die Omni-Wheels an der Motorwelle zu montieren, wird ein Adapter konstruiert, der die Durchmesserunterschiede ausgleicht.

Die Halterungen, Abstandhalter und Adapter sind aus Aluminium gefertigt, um ein zu hohes Gewicht des Aufbaus zu vermeiden.

Programmierung und Elektronik

Die theoretischen Grundlagen des Balanciervorgangs basieren hauptsächlich auf Berechnungen in den Themengebieten der Mehrkörpersysteme und der Regelungstechnik.

Der Softwareaufbau eines Arduino-Boards ist grundsätzlich in zwei Teile aufgeteilt. Im ersten Teil, dem Setup, werden Befehle eingefügt, die nur einmal zu Beginn durchlaufen werden müssen.

Der zweite Teil des Aufbaus besitzt die Funktion einer Schleife und wird vorwiegend als Hauptteil verwendet. Dieser Abschnitt wird stetig wiederholt und arbeitet dabei die Befehle in der programmierten Reihenfolge ab. Allgemein ist das Programm des Roboters in drei Schritte zu unterteilen: die Messdatenaufnahme, die Berechnung der Motordaten sowie die Kommunikation und Datenübertragung.

Die Schleife ist so programmiert, dass sie alle 20 Millisekunden wiederholt wird. Hierbei werden zu Beginn die vom Sensor gemessenen Werte der Winkelabweichung und der Winkelgeschwindigkeit ausgelesen. Das Programm auf dem Arduino-Mega durchläuft nun den Berechnungsteil (siehe Abb. 2). Hierbei werden die gemessenen Werte umgerechnet, sodass für jeden Schrittmotor die zum Balancieren notwendigen Geschwindigkeitskomponenten bestimmt werden.

```
SensorInput(); // Sensor data input
CalcWheelControl(); // Calculate Wheelspeed
timebefore = millis();
DataTransfer(); // Transfer Wheelspeed data to Arduino Uno's
SaveSpeedData(); // Save old and new Speed value
CalcBallPosSpeed(); // Calculate Ball Position and Ball Speed
```

Abbildung 2: Berechnungsablauf

Aufgrund der Eigenschaft eines Arduino-Boards, die Befehle der Schleife nacheinander zu befolgen, können bei der Verwendung eines einzelnen Boards die Motoren nur nacheinander angesteuert werden. Daher wird jeder Motor an einen eigenen Arduino-Uno mit Motor-Shield angeschlossen. Die Daten der Motorgeschwindigkeiten können so im Millisekundenbereich nacheinander vom Arduino-Mega zu den einzelnen Arduino-Unos gesendet werden, sodass ein nahezu gleichzeitiges Ansteuern der Motoren möglich ist. Der empfangene Wert wird ausgelesen und über das Motor-Shield an den Motor weitergegeben. Durch einen kurzen Programmaufbau wird der Rechenaufwand reduziert.

Die Kommunikation zwischen dem Hauptboard Arduino-Mega und den kleineren Motorboards Arduino-Uno erfolgt über die serielle Schnittstelle und eine Verbindung über einen digitalen Pin. Über den digitalen Pin wird ein Signal (HIGH) vom Arduino-Mega zum Arduino-Uno übertragen. Direkt danach wird der Wert der Motorgeschwindigkeit über die serielle Schnittstelle übermittelt. Der Uno empfängt zuerst das Signal am digitalen Pin und unterbricht danach aufgrund eines „Interrupt“-Befehls die Schleife. Nun wird eine extra angelegte Funktion aufgerufen, welche den empfangenen Wert an der seriellen Schnittstelle ausliest und einer Variablen zuordnet (siehe Abb. 3). Anschließend wird der Pin wieder auf LOW geschaltet und die Motorgeschwindigkeit vom Arduino-Mega neu berechnet. Die dafür notwendigen Sensordaten werden alle 20 Millisekunden vom Arduino-9-Axes-Motion-Shield übertragen.

```
attachInterrupt(digitalPinToInterrupt(pin),reading,RISING);
...
void reading(){
  if( Serial.available() > 1){
    speed_1 = Serial.parseFloat();
    Serial.println(speed_1);
  }
}
```

Abbildung 3: Interrupt und Wertzuweisung

Schaltplan

Um den elektrischen Aufbau und die dazugehörige Verkabelung der einzelnen Komponenten zu verdeutlichen, ist nachfolgend der Schaltplan des Roboters dargestellt.

Zur Spannungsversorgung des Roboters wird ein Labornetzgerät mit einer Ausgabespannung von 12V verwendet. Über eine Verteilerdose werden der Arduino-Mega und das Adafruit-Motorshield mit Spannung (12V) versorgt. Die Schrittmotoren und Arduino-Unos sind mit den jeweiligen Motorshields verbunden und erhalten über diese die erforderliche Betriebsspannung.

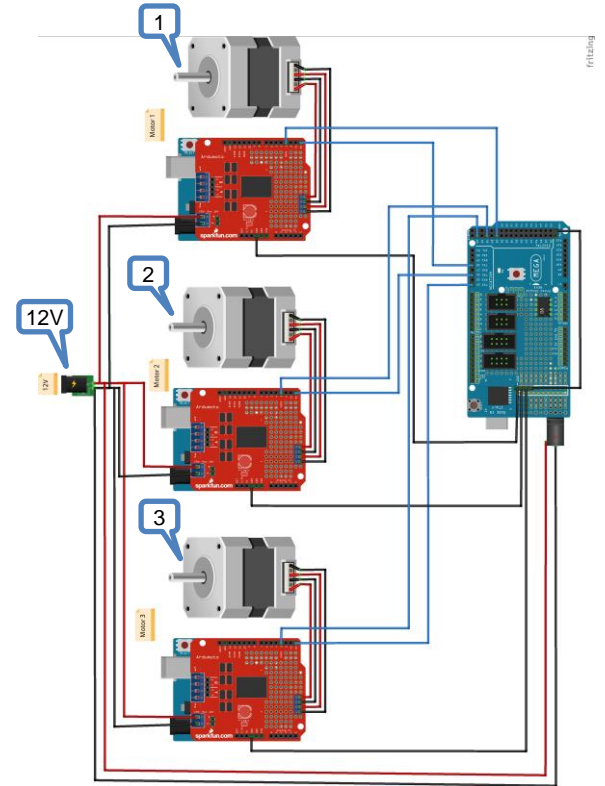


Abbildung 4: Schaltplan des Roboters

Versuche und Durchführung

Für die Erprobung der Motoransteuerung und der Kommunikation zwischen zwei Arduino-Boards werden mehrere Testversuche durchgeführt. Ziel dabei ist es, eine sichere Kommunikation vom Arduino-Uno über das Motor-Shield zum Schrittmotor herzustellen. Außerdem ist ein sicherer Datentransfer zwischen dem Arduino-Mega und den drei Arduino-Unos zu gewährleisten, um eine Fehlfunktion der Schrittmotoren zu verhindern. Die Arduino-Software stellt einige Beispielprogramme zur Verfügung, die zur Lösungsfindung eingesetzt werden.

In weiteren Versuchen werden die Drehrichtungen der Schrittmotoren in Abhängigkeit der gesendeten Daten des Bewegungssensors ermittelt und angepasst. Um einen korrekten Balancierablauf sicherzustellen, werden weitere Versuche mit unterschiedlichen Regelparametern am aktuellen Aufbau des Roboters durchgeführt.

Ergebnisse und Ausblick

Innerhalb des IFPs ist es gelungen, einen Roboter vom anfänglichen CAD-Modell zum fertigen Systemaufbau zu realisieren. Des Weiteren ist eine zentrale Stromversorgung installiert worden, über die ein Netzgerät den Roboter mit Energie versorgt. Die elektrischen Verbindungen der einzelnen Komponenten sowie ein funktionierendes Programm konnten ebenfalls erstellt werden. Das eigenständige Balancieren des Roboters auf einem Ball konnte nicht vollendet werden, weil eine Auswahl der richtigen Regelparameter nicht getroffen werden konnte.

Dieser Schritt ist in weiteren Forschungsprojekten durchzuführen. Außerdem ist eine mobile Stromversorgung über geeignete Akkus vorzunehmen, sodass die Verwendung eines Netzteils entfällt. Das Anschließen einer Fernbedienung stellt eine mögliche Ausbaustufe dar, um den Roboter gezielt zu steuern. Außerdem können weitere Sensoren und Kameras an den Roboter angeschlossen und in die Steuerung integriert werden, um ein autonomes Fahren zu ermöglichen.

Kontakt

Prof. Dr.-Ing. Dietmar Jennewein
Fachbereich Maschinenbau und Kunststofftechnik
Schöfferstraße 3, 64295 Darmstadt
E-Mail: dietmar.jennewein@h-da.de