Entwurfsmethodik
Institut für Informatik

JOHANN WOLFGANG GOETHE
UNIVERSITÄT
FRANKFURT AM MAIN

Masterarbeit

Auf dem Weg zum autonom fahrenden Fahrrad: Eine videogestützte Regelung des Weges mit Webbots

Amine Karabila

Wintersemester 2025

Abstract

Diese Masterarbeit befasst sich mit der Simulation eines selbstbalancierenden Fahrrads unter Verwendung des Simulationsprogramms V-REP (VBOTS) in Kombination mit MATLAB. Ziel ist es, zwei Regelungsansätze zu implementieren und zu analysieren:

- Ein PID-Regler, basierend auf der Arbeit eines Bachelorabsolventen, der klassische Regelungstechniken zur Stabilisierung nutzt.
- Ein modellprädiktiver Regler (Model Predictive Control, MPC), der auf Kameradaten eines YOLO-Modells zurückgreift, um Fahrbahnwege zu erkennen und die Fahrtrichtung aus Videobildern zu extrahieren.

Die Kombination dieser beiden Regelungen zielt darauf ab, die Vorteile beider Ansätze zu nutzen und eine möglichst realistische Simulation des selbstbalancierenden Fahrrads zu erreichen. Durch die Integration von Bildverarbeitung und fortschrittlicher Regelungstechnik soll ein Beitrag zur Weiterentwicklung autonomer Zweiradsysteme geleistet werden.

This master's thesis focuses on the simulation of a self-balancing bicycle using the simulation program V-REP (VBOTS) in combination with MATLAB. The aim is to implement and analyze two control approaches:

- A PID controller, based on the work of a bachelor's graduate, utilizing classical control techniques for stabilization.
- A model predictive controller (Model Predictive Control, MPC) that leverages camera data from a YOLO model to detect bike paths and extract the riding direction from video images.

The combination of these two control strategies aims to harness the advantages of both approaches to achieve a highly realistic simulation of the self-balancing bicycle. By integrating image processing and advanced control techniques, this work seeks to contribute to the advancement of autonomous two-wheeled systems.

Inhaltsverzeichnis

1 Einleitung

Die Entwicklung selbstbalancierender Fahrräder stellt eine anspruchsvolle Aufgabe im Bereich der Regelungstechnik dar, da sie die Stabilisierung eines instabilen Systems in Echtzeit erfordert. Ziel dieser Arbeit ist die realitätsnahe Simulation eines solchen Fahrrads mithilfe des Simulationsprogramms V-REP (VBOTS) in Kombination mit MATLAB. Dabei werden zwei unterschiedliche Regelungsansätze implementiert und analysiert:

- Ein PID-Regler, basierend auf der Arbeit eines Bachelorabsolventen, der klassische Regelungstechniken zur Stabilisierung nutzt.
- Ein modellprädiktiver Regler (Model Predictive Control, MPC), der auf Kameradaten eines YOLO-Modells zurückgreift, um Fahrbahnwege zu erkennen und die Fahrtrichtung aus Videobildern zu extrahieren.

Die Kombination dieser beiden Regelungen zielt darauf ab, die Vorteile beider Ansätze zu nutzen und eine möglichst realistische Simulation des selbstbalancierenden Fahrrads zu erreichen. Durch die Integration von Bildverarbeitung und fortschrittlicher Regelungstechnik soll ein Beitrag zur Weiterentwicklung autonomer Zweiradsysteme geleistet werden.

2 Webots – Aufbau, Funktionsweise und Einsatz im Projektkontext

Die Open-Source-Software Webots ist ein Simulationsprogramm zur realitätsnahen Modellierung und Steuerung von Robotern in virtuellen 3D-Umgebungen. Webots wird in Industrie, Lehre und Forschung eingesetzt und bietet eine vollständige Entwicklungsumgebung, um Roboter sowie deren Umgebung zu modellieren, zu programmieren und zu simulieren. Entwickelt wurde Webots ursprünglich 1996 am EPFL und wird seit 1998 von Cyberbotics kontinuierlich weiterentwickelt. Im Dezember 2018 wurde Webots unter der Apache-2.0 Lizenz als Open Source freigegeben, wodurch es eine aktive Entwicklergemeinschaft mit regelmäßigen Beiträgen und Aktualisierungen gibt (über 100 Mitwirkende auf GitHub). Im Folgenden werden die Grundstruktur von Webots, das Open-Source-Projekt, die API-Schnittstellen (insbesondere zu MATLAB und ROS) sowie spezifische Aspekte wie Kamerasensorik, Physik-Engine und Echtzeitfähigkeit erläutert. Dabei wird jeweils der Kontext eines selbstbalancierenden, autonom fahrenden Fahrrads hervorgehoben, um die Relevanz für das vorliegende Projekt deutlich zu machen.

2.1 Grundlegende Merkmale und Struktur von Webots

Ein Webots-Simulationsprojekt besteht aus mehreren zentralen Komponenten. Welt-Dateien (Dateiendung .wbt) definieren die virtuelle Umgebung samt [web(2024)] Roboterinstanzen und Objekten. In einer solchen World werden alle Objekte hierarchisch beschrieben, inklusive ihrer Position, Orientierung, Geometrie, visuellen Eigenschaften (Farbe, Textur) und physikalischen Parameter (Masse, Reibung, etc.). Eine Welt kann mehrere Roboter enthalten und referenziert deren jeweilige Controller-Programme – sie enthält jedoch nicht den Quellcode der Controller selbst, sondern nur den Namen des zu startenden Controllers je Roboter. Webots-Weltdateien verwenden ein textbasiertes Format (früher VRML-ähnlich, seit R2025a auf W3D/X3D basierend), das bei Bedarf externe Robotermodelle einbindet. Solche Robotermodelle werden häufig als PROTO-Dateien (.proto) realisiert. Eine PROTO-Datei ist ein wiederverwendbarer Vorlagendefinitionsdatei für Roboter oder Objekte mit parametrierbaren Eigenschaften. Im .wbt-Szenario kann man also statt eines vollständigen Roboters nur eine PROTO-Instanz referenzieren – das erlaubt eine modulare Strukturierung. Die Weltdatei kann auf mehrere PROTOs und Ressourcendateien (Texturen, Meshes) zurückgreifen, was die Simulationsszenen flexibel erweiterbar macht.

Jeder Roboter in Webots wird durch einen eigenen Controller-Prozess gesteuert. Ein Controller ist ein Programm, das die Sensoren ausliest und Aktuatoren des Roboters steuert, Webots unterstützt hierfür mehrere Programmiersprachen: C, C++, Python, Java sowie MATLAB®. Beim Start der Simulation wird für jeden Roboterprozess der angegebene Controller gestartet und mit dem entsprechenden Roboternode verknüpft. Mehrere Roboter können zwar dasselbe Controller-Programm nutzen, laufen aber jeweils in separaten Prozessen, um parallele Abläufe zu ermöglichen. Die Wahl der Programmiersprache beeinflusst die Handhabung: C/C++ Controller werden vorab zu nativen Binärprogrammen kompiliert, während Python- und MATLAB-Controller zur Laufzeit von den jeweiligen Interpreterumgebungen ausgeführt werden. Java-Controller stellen einen Sonderfall dar (Kompilierung zu Bytecode und Ausführung in der JVM). In der Praxis bedeutet dies, dass Webots z.B. einen Python-Controller mit dem installierten Python-Interpreter startet oder einen MATLAB-Controller über eine MATLAB-Engine ausführt. Im Kontext des selbstbalancierenden Fahrrads wurde diese Flexibilität genutzt, indem Regelungsalgorithmen teils in MATLAB entwickelt und direkt als Webots-Controller eingebunden wurden. So kann ein PID- oder MPC-Regler im MATLAB-Skript die Balance des simulierten Fahrrads steuern, während Webots die Physiksimulation und Sensorik (z.B. Gyro- und Beschleunigungssensoren am Fahrradmodell) bereitstellt.

Webots stellt für das Modellieren und Simulieren eine moderne grafische Benutzeroberfläche zur Verfügung, mit der sich Roboter, Objekte und Umgebungen interaktiv erstellen und modifizieren lassen. Die Software liefert eine umfangreiche Bibliothek von vordefinierten Robotern, Sensoren, Aktuatoren und Objekten, die über eine Szene-Baum-Struktur verwaltet werden. So sind z.B. gängige Forschungsroboter (e-puck, KUKA YouBot, NAO usw.) als PROTO-Modelle bereits vorhanden, was den Einstieg erleichtert. Für eigene Roboter – wie das autonome Fahrrad – kann man entweder ein passendes bestehendes Modell adaptieren oder mittels PROTO und Basis-Nodes (Solid, Joint, Shape etc.) ein eigenes dynamisches Modell aufbauen. Die Standard-Projektstruktur von Webots sieht vor, dass alle zugehörigen Dateien in einem Projektordner organisiert sind (typischerweise mit Unterordnern worlds/ für Weltdateien, protos/ für PROTO-Definitionen, controllers/ für Controller-Code usw.). Dadurch bleiben komplexe Simulationen übersichtlich und reproduzierbar.

Zusammenfassend bietet Webots alle nötigen Bausteine, um ein komplexes mechatronisches System wie ein selbstfahrendes Fahrrad virtuell abzubilden: Die Weltdatei definiert Gelände und Startbedingungen, PROTO-Modelle beschreiben die Mechanik (Rahmen, Räder, Lenkung, Sensoren) und Controller implementieren die Regelungslogik, wobei Webots als Infrastruktur die Sensor-Aktorkopplung in Echtzeit und eine grafische 3D-Visualisierung sicherstellt.

2.2 Webots als Open-Source-Projekt (GitHub-Repositorium)

Literatur

[web(2024)] Webots – github repository. https://github.com/cyberbotics/webots, 2024. Allgemeine Projektübersicht.