

Fahrbarer Brückenleger

Amr Khaled Mohamed Bedeir, Elektro- und Informationstechnik
Otto-von-Guericke-Universität Magdeburg

Zusammenfassung—Im Rahmen des Projektseminars 2026 wurde unter Verwendung der LEGO-Mindstorms-Plattform ein autonomes Robotersystem mit dem offiziellen Namen „Fahrbarer Brückenleger“ entwickelt. Ziel war es, ein System zu konstruieren, das selbstständig Bodenlücken erkennt und diese durch das Verlegen einer mobilen Brücke überwindet. Die Softwaresteuerung und die Implementierung der Logik wurden vollständig in MATLAB realisiert. Zur Entfernungsmessung wurde ein Ultraschallsensor eingesetzt. Zwei große Motoren dienten zur Bewegung des Roboters, während mittelgroße Servomotoren für das Absenken der Brücke verwendet wurden. Die Funktionsfähigkeit wurde in mehreren Testläufen erfolgreich nachgewiesen, wobei eine präzise Positionierung, ein stabiler Verlegevorgang sowie eine sichere Überquerung dokumentiert wurden. In diesem Paper werden das mechanische Design, die sensorische Datenverarbeitung sowie die softwareseitigen Herausforderungen detailliert erläutert.

Schlagwörter—Skorpion, MATLAB, Große Motoren, Ultraschallsensorik.

I. EINLEITUNG

AUTONOME Robotersysteme spielen in der modernen Technik eine entscheidende Rolle, insbesondere bei der Navigation in komplexem oder unterbrochenem Gelände. In Bereichen wie der Katastrophenhilfe und Infrastrukturentwicklung ist die Fähigkeit eines Systems, geografische Hindernisse selbstständig zu überwinden, von großer Bedeutung. Im Rahmen des Projektseminars Elektrotechnik und Informationstechnik 2026 an der Otto-von-Guericke-Universität Magdeburg wurden diese technischen Prinzipien im Modellmaßstab umgesetzt.

Das Hauptziel dieses Projekts war die Entwicklung eines autonomen Brückenbauroboters. Das System wurde so konzipiert, dass es mithilfe von Sensortechnologie selbstständig Lücken im Boden erkennt und dann eine mobile Brückenkonstruktion einsetzt, um eine sichere Überquerung zu ermöglichen. Das mechanische Design des Roboters wurde von der Form eines Skorpions [1] inspiriert (siehe Abbildung 1). Für die mechanische Umsetzung wurde das LEGO-Mindstorms-EV3-System gewählt, während die Steuerungslogik und die Sensordatenverarbeitung in MATLAB [2] implementiert wurden.

Dieser Artikel dokumentiert den gesamten Entwicklungsprozess, beginnend mit dem mechanischen Design und der Sensorauswahl. Darüber hinaus wird die algorithmische Umsetzung in der MATLAB-Umgebung detailliert beschrieben. Abschließend werden die Ergebnisse der Funktionstests kritisch bewertet.

II. VORBETRACHTUNGEN

In diesem Abschnitt werden die technischen Spezifikationen der verwendeten Hardwarekomponenten sowie die Softwareumgebung für den Brückenleger-Roboter detailliert beschrieben.



Abbildung 1. Skorpion [1]



Abbildung 2. Fertiger „Fahrbarer Brückenleger“

A. LEGO-Mindstorms-EV3-Hardware

Als modulare Hardware-Plattform wurde das LEGO-Mindstorms-EV3-System gewählt. Für die mechanische Umsetzung der Bewegungsabläufe wurde eine Kombination aus großen und mittleren Servomotoren verwendet:

Große Motoren: Diese Einheiten [3] weisen ein hohes Drehmoment auf und wurden für den primären Antrieb des Fahrzeugs eingesetzt.

Mittlere Motoren: Aufgrund der kompakten Bauweise wurden diese für das Auslegen der Brückenkonstruktion und für sekundäre Mechanismen genutzt, die eine höhere Platzeffizienz erforderten.

B. MATLAB-Entwicklungsumgebung

Die Implementierung der Steuerungslogik erfolgte vollständig in MATLAB [2]. Über eine spezialisierte Toolbox wurde eine Echtzeit-Kommunikation zwischen dem EV3-Stein und der Workstation hergestellt, um Sensordaten zu verarbeiten und Motorbefehle synchron auszuführen.

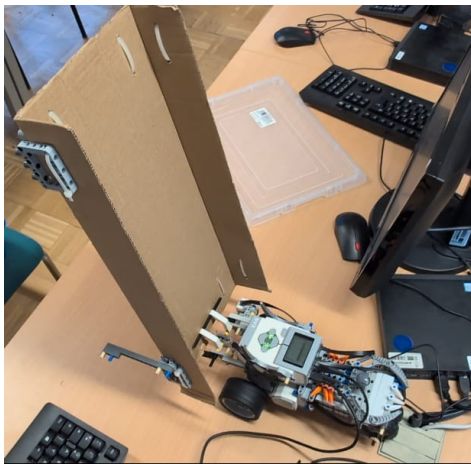


Abbildung 3. Oberer Winkel, während der Roboter die Brücke hält

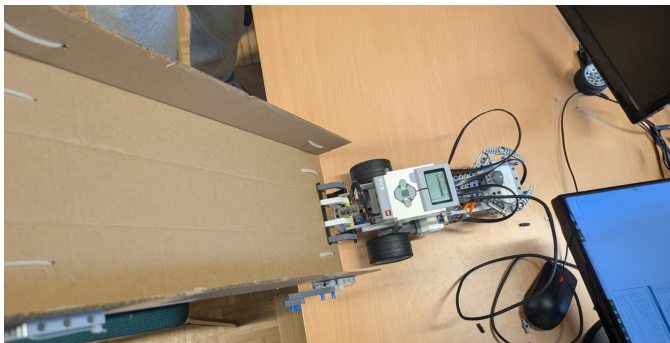


Abbildung 4. Detailaufnahme des Roboters beim Brückenverlegevorgang

C. Ultraschallsensorik

Zur autonomen Erkennung von Bodenlücken wurde ein EV3-Ultraschallsensor [4] eingesetzt. Der Sensor misst die Zeit t , die das Schallecho für den Rückweg benötigt. Basierend auf der Schallgeschwindigkeit wird die Distanz d zum Boden berechnet. Ein signifikanter Anstieg des Wertes d signalisiert eine Bodenlücke.

III. KONSTRUKTION UND PROGRAMMIERUNG

Die Entwicklung des Brückenleger-Roboters ist durch eine synergetische Integration mechanischer Komponenten und einer in MATLAB [2] implementierten übergeordneten Steuerungsstruktur gekennzeichnet.

A. Mechanisches Design und Aktuatorik

Die physische Architektur wurde so konzipiert, dass sie den dynamischen Kräften während der Brückenverlegungsphase standhält (siehe Abbildung 4).

Antriebssystem: Der Antrieb wurde über zwei große EV3-Servomotoren [3] realisiert, die in einer Hinterradantriebs-Konfiguration integriert wurden. Diese ermöglicht eine differenzielle Lenkung, die genutzt wurde, um eine präzise Ausrichtung während der Annäherung an eine Bodenlücke beizubehalten.

Verlegemechanismus: Ein dritter Motor wurde ausschließlich für die Betätigung des Brückenlegearms eingesetzt. Ein

Getriebesystem mit hoher Übersetzung wurde implementiert, um die schnelle Rotation des Motors in ein hohes Drehmoment umzuwandeln und so ein stabiles Absenken der Brücke zu gewährleisten.

Chassis-Stabilität: Um den nach vorne verschobenen Schwerpunkt während der Verlegung auszugleichen, wurde der EV3-Stein am Heck des Fahrzeugs als Gegengewicht positioniert. Die finale mechanische Montage ist in Abbildung 2 und ergänzend in Abbildung 3 dargestellt.

B. Sensorintegration und Logikarchitektur

Das autonome Verhalten des Systems wird durch einen zustandsbasierten Regelkreis gesteuert (siehe Abbildung 5). Während der Navigation überwacht der Ultraschallsensor [4] kontinuierlich den Abstand d zum Boden. Ein Bodenabstand wird erkannt, wenn der gemessene Abstand $d = X$ cm einen vordefinierten Schwellenwert $d > (X + 3)$ cm überschreitet, was auf einen Verlust der Oberflächenreflexion hinweist. Bei Erkennung werden die Antriebsmotoren sofort abgebremst.

C. MATLAB-Implementierung

Die Steuerungsalgorithmen wurden als Echtzeit-Skript in MATLAB [2] entwickelt, wobei die spezialisierte EV3-Toolbox zur Kommunikation genutzt wurde.

1) *Initialisierung und Kalibrierung:* Um Fehlfunktionen zu vermeiden, wurde das System durch Löschen aktiver Variablen initialisiert. Es wurde eine Kalibrierungsphase durchgeführt, in der der Ultraschallsensor [4] die Bodenfreiheit misst. Ein Sicherheitsschwellenwert wird berechnet, indem 3 cm zum Ausgangswert addiert werden, um einen zuverlässigen Auslösepunkt für die Spaltdetektion zu schaffen.

2) *Lückensuche und Kantenerkennung:* Der Roboter hält eine gerade Bahn, indem er mit einer konstanten Leistung von 20% der Leistung der großen Motoren [3] fährt. Während der Fahrt überwacht das System kontinuierlich den Abstand zum Boden. Wenn dieser Wert plötzlich über den Sicherheitsgrenzwert $d > (X + 3)$ cm steigt, was darauf hinweist, dass der Roboter eine Lücke erreicht hat, löst die Steuerungslogik einen sofortigen Bremsbefehl aus. Dieser Schnellstopp sorgt dafür, dass der Roboter am Rand der Öffnung stabil bleibt, ohne zu fallen oder den Kontakt zum Boden zu verlieren.

3) *Ausrichtung und Brückenverlegung:* Da der Sensor an der Vorderseite angebracht ist, bewegt sich der Roboter während einer Ausrichtungsphase 0,4 s lang rückwärts, um den Brückenarm 5 cm vom Rand entfernt zu positionieren. Anschließend wird der mittelgroße Servomotor mit einem Grenzwert von 1800° aktiviert. Eine tachometerbasierte Steuerung wurde einer zeitbasierten Steuerung vorgezogen, um trotz Schwankungen der Batteriespannung eine gleichbleibende Präzision zu gewährleisten.

4) *Sicherung und parallele Ausführung:* Ein Einhakmanöver wird durchgeführt, indem 2 s lang rückwärts gefahren wird, um die Brücke zu sichern. Dadurch werden Unfälle verhindert, die durch Reibung zwischen dem Brückenarm und der Brücke selbst verursacht werden könnten. Anschließend hebt der Roboter seinen Arm um 900° an, was der Hälfte des gesamten Bewegungsbereichs entspricht. Dieser Schritt stellt sicher,

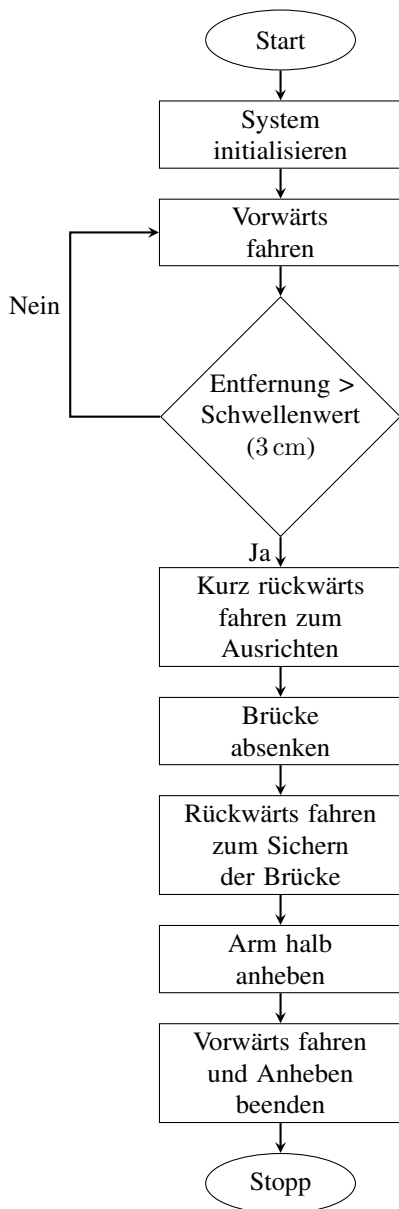


Abbildung 5. Flussdiagramm des fahrbaren Brückenlegers

dass der Arm nicht mit der installierten Brücke kollidiert, während der Roboter seine nächste Bewegung ausführt. Um die Effizienz zu verbessern, fährt der Roboter 5 s lang vorwärts, hebt gleichzeitig die zweite Hälfte seines Arms an und passiert die Brücke. Der Vorgang endet mit einem akustischen Signal und der Beendigung der Hardwareverbindung.

IV. ERGEBNISSE UND DISKUSSION

A. Ergebnis

Insgesamt wurden die Projektziele vollständig erreicht, da das sichere und autonome Überqueren von Bodenspalten im Rahmen des LEGO-Mindstorms-Projekts wiederholt demonstriert werden konnte. Der Roboter identifizierte den Zielbereich erfolgreich und bewältigte den Übergangsprozess zuverlässig. Dies bestätigt, dass sowohl die entwickelte Logik als auch die

strukturellen Anpassungen für die spezifischen Anforderungen der Aufgabenstellung effektiv waren.

B. Probleme – Strukturelle und sensorische Einschränkungen

1) *Strukturelle Herausforderungen und Stabilität:* Die größte Schwierigkeit während der Entwicklung bestand darin, dass die Brücke extrem schwer und überdimensioniert war. Infolgedessen verfügte der ursprüngliche Hebearm nicht über die erforderliche Kraft, und die gesamte Roboterkonstruktion war instabil. Um dieses Problem zu lösen, wurde eine Untersetzung implementiert, um das Drehmoment des Arms zu erhöhen, und die Größe der Halterung wurde um das Dreifache vergrößert (siehe Abbildung 6), wobei zwei flexible Arme hinzugefügt wurden, die sich gleichzeitig mit der Brücke bewegen, um wie eine Treppe zu wirken und jegliches Herunterfallen auf die Brücke zu verhindern (siehe Abbildung 7).

Darüber hinaus wurde die gesamte Form des Roboters modifiziert, um das hohe Gewicht der Brücke besser aufzunehmen und eine stabilere, widerstandsfähigere Plattform zu gewährleisten.

2) *Sensorplatzierung und Näherungslogik:* Eine große Herausforderung betraf die physische Positionierung des Ultraschallsensors [4] in Bezug auf den Radstand des Roboters. Die direkte Montage des Sensors an der Vorder- oder Rückseite führte zunächst zu Fehlern: Aufgrund des geringen Abstands zwischen dem Erfassungspunkt und dem Chassis fiel der Roboter oft über die Kante, bevor ein Befehl für die Rückwärtsbewegung ausgeführt werden konnte.

Um dieses Problem zu beheben, wurde die Chassis-Verlängerung des Roboters vergrößert, um einen Abstand von mindestens 2 cm zwischen dem Sensor und dem Hauptkörper zu schaffen. Dieser Puffer ermöglichte es dem Sensor, den Abgrund rechtzeitig zu erkennen und den Reset zu initiieren, bevor der Roboter den Kontakt zum Boden verlor. Darüber hinaus wurde die vertikale Höhe des Sensors präzise kalibriert, um eine zuverlässige Erkennung zu gewährleisten, ohne dass der Sensor auf dem Boden schleift oder als physisches Hindernis für die Brücke wirkt.

V. FAZIT

Zusammenfassend lässt sich feststellen, dass die Entwicklung des autonomen Roboters zum Überqueren von Bodenlücken im Rahmen des LEGO-Mindstorms-Projekts erfolgreich abgeschlossen wurde. Trotz der erheblichen mechanischen Herausforderungen, die sich aufgrund des hohen Gewichts und der Größe der Brücke ergaben, konnten durch gezielte Hardware-Modifikationen funktionale Lösungen umgesetzt werden.

Durch den Einbau eines Untersetzungsgetriebes und die strukturelle Neugestaltung des Fahrgestells wurde sichergestellt, dass der Roboter über die erforderliche Stabilität und Kraft für den Brückenmechanismus verfügt. Darüber hinaus ermöglichte die präzise Kalibrierung der Sensorposition, insbesondere durch Einhaltung eines Abstands von 2 cm zum Chassis, eine zuverlässige Spaltenerkennung und verhinderte kritische Fahrfehler an den Rändern des Bodens.

Das Projekt zeigt erfolgreich, dass eine robuste autonome Steuerung nur durch das optimale Zusammenspiel von mechanischer Belastbarkeit und präziser Sensortechnik erreicht

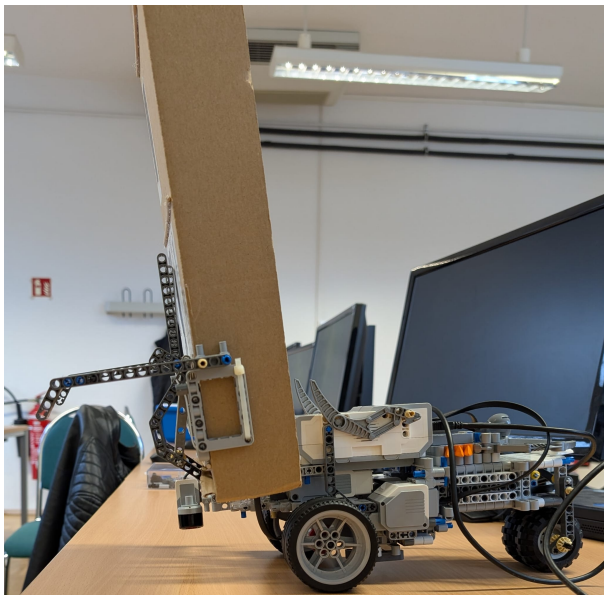


Abbildung 6. Mehr Details zur Brückenform

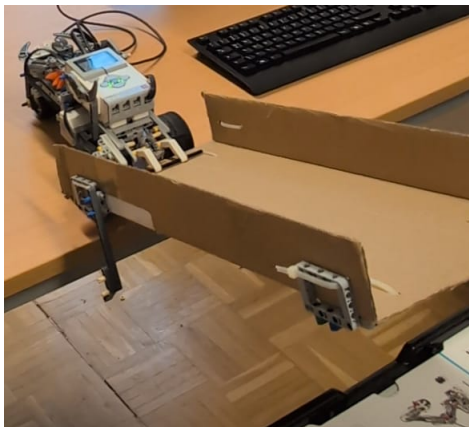


Abbildung 7. Brückenwinkel für den Installationsprozess

werden kann. Alle Projektziele wurden vollständig erreicht, und die Funktionalität des Systems wurde durch wiederholte Testläufe erfolgreich nachgewiesen.

VI. ANHANG

Der in Listing 1 gezeigte Auszug enthält die Hauptlogik des Steuerungsskripts.

Listing 1. Kernalgorithmus zur Spalterkennung und Brückenlegung

```
% --- HAUPTLOGIK: SPALTERKENNUNG ---
fwdPower = 20; % Reduzierte Leistung fuer
    Praezision
mLeft.power = fwdPower;
mLeft.syncedStart(mRight);

while true
if uSensor.value > gapThreshold
mLeft.stop();
mRight.stop();
fprintf('Kante erkannt!\n');
break;
end
```

```
end
```

```
% --- BRUECKENMANOEVER ---
% Ausrichtung des Arms (0.4s Rueckwaertsfahrt)
mLeft.power = -25;
mLeft.syncedStart(mRight)
pause(0.4);
mLeft.stop();

% Absenken der Bruecke (1800 Grad Rotation)
mBridge.setProperties('power', -15, 'limitMode'
    ', 'Tacho', 'limitValue', 1800);
mBridge.start();
mBridge.waitFor();

% Fixierung durch kurzes Zuruecksetzen
mLeft.power = -25;
mLeft.syncedStart(mRight);
pause(2.0);
mLeft.stop();

% --- UEBERQUERUNG UND PARALLELES HEBEN ---
mBridge.setProperties('power', 15, 'limitMode'
    ', 'Tacho', 'limitValue', 900);
mBridge.start(); % Hebevorgang starten

mLeft.power = 35;
mLeft.syncedStart(mRight); % Gleichzeitiges
    Losfahren
pause(5);

mLeft.stop();
brick.beep();
```

LITERATURVERZEICHNIS

- [1] A-Z Animals, "Scorpion," [Online]. Available: <https://a-z-animals.com/animals/scorpion/>
- [2] MathWorks, "MATLAB Support Package for LEGO MINDSTORMS EV3 Hardware," 2026. [Online]. Available: <https://www.mathworks.com/hardware-support/lego-mindstorms-ev3-matlab.html>
- [3] Brick Owl, "LEGO Mindstorms EV3 Large Motor 95658," [Online]. Available: <https://www.brickowl.com/catalog/lego-mindstorms-ev3-large-motor-95658>
- [4] Raising Robots, "EV3 Ultrasonic Sensor," [Online]. Available: <https://raisingrobots.com/product/ev3-ultrasonic-sensor/>