

Konstruktion und Programmierung eines autonomen Robo-Transporters zur Lastenmanipulation

Bohdan Soldatko, Elektrotechnik und Informationstechnik
Otto-von-Guericke-Universität Magdeburg

Zusammenfassung—Das vorliegende Paper beschreibt die Entwicklung und Konstruktion eines autonomen Transportsystems unter Verwendung von LEGO Mindstorms EV3 und MATLAB. Im Fokus der Arbeit steht die mechanische Realisierung eines stabilen Hubmechanismus in Form eines Gabelstaplers sowie die softwareseitige Implementierung einer präzisen Steuerung. Das System ist in der Lage, Objekte mittels Ultraschallsensoren zu detektieren und autonom aufzunehmen. Die Kommunikation zwischen der Steuerungssoftware und der Hardware wird über Bluetooth realisiert. Die Ergebnisse der Erprobungsphase werden detailliert dargelegt und mögliche Optimierungspotenziale für zukünftige Weiterentwicklungen aufgezeigt.

Schlagwörter—Greifarm, Transporter, Manipulator, Ultraschallsensor, Gabelstapler

I. EINLEITUNG

AUTONOME Transportsysteme und fahrerlose Transportfahrzeuge (FTF) stellen ein zentrales Forschungsfeld in der modernen Intralogistik und Automatisierungstechnik dar [1]. Sie werden eingesetzt, um Materialflüsse effizienter zu gestalten und menschliche Fehlerquellen zu minimieren. Entsprechende Prototypen und Machbarkeitsstudien lassen sich im Labormaßstab hervorragend mit modularer Robotik abbilden.

Im Rahmen dieses Projekts wurde ein Prototyp entwickelt, der die grundlegenden Prinzipien der automatisierten Lastenmanipulation demonstriert. Die zentrale Herausforderung bestand in der Konstruktion eines Roboters, der nicht nur eine ausreichende mechanische Stabilität für Hebevorgänge aufweist, sondern auch über eine zuverlässige Umgebungserfassung verfügt. Das System wurde so konzipiert, dass es sowohl teleoperiert über eine grafische Benutzeroberfläche als auch in teilautonomen Routinen betrieben werden kann [2].

Der folgende Bericht dokumentiert die systematische Entwicklung, von den grundlegenden Anforderungen über den mechanischen Aufbau bis hin zur Programmierung der Algorithmen.

II. SYSTEMANFORDERUNGEN UND VORÜBERLEGUNGEN

Dieser Abschnitt definiert die grundlegenden mechanischen und softwareseitigen Anforderungen, die das spätere Designkonzept des Roboters maßgeblich bestimmen. Dabei wurden insbesondere die physikalischen Grenzen der verwendeten LEGO-Komponenten berücksichtigt.

A. Anforderungen an die Mechanik

Die grundlegenden Anforderungen an die Robotik-Hardware bestanden in der Fähigkeit, eine Nutzlast von bis zu 0,35 kg stabil anzuheben und zu transportieren, ohne dass das System das Gleichgewicht verliert. Hierfür war ein tiefer Schwerpunkt essenziell. Da die Last an der Front des Roboters aufgenommen wird, verlagert sich der Gesamtschwerpunkt bei Beladung signifikant nach vorne. Diesem Effekt musste durch ein ausreichendes Gegengewicht im Heckbereich sowie einen verlängerten Radstand entgegengewirkt werden. Zudem musste das Fahrwerk so ausgelegt werden, dass eine hohe Traktion auf verschiedenen Untergründen gewährleistet ist, während gleichzeitig eine hohe Wendigkeit für das Manövrieren auf engem Raum erhalten bleibt.

B. Anforderungen an Sensorik und Software

Um Objekte vor dem Roboter zuverlässig zu erfassen, wurde eine frontal ausgerichtete Sensorik gefordert. Das System muss Entfernungen präzise messen können, um den Greifmechanismus im exakt richtigen Moment auszulösen. Eine Fehlmessung in diesem Bereich würde dazu führen, dass die Gabeln die Last entweder rammen oder ins Leere greifen. Softwareseitig wurde eine Implementierung in MATLAB vorgegeben, welche eine benutzerfreundliche Schnittstelle zur Überwachung und Steuerung (App Designer) sowie stabile Kommunikationsprotokolle zur Datenübertragung an den Microcontroller erfordert [3]. Die Latenzzeit der Bluetooth-Verbindung durfte dabei einen kritischen Schwellwert nicht überschreiten, um Echtzeitreaktionen zu ermöglichen.

III. TECHNISCHE UMSETZUNG UND HARDWARE

Die konstruktive Umsetzung gliedert sich im Wesentlichen in den Aufbau des mobilen Fahrwerks, die mechanische Realisierung des Hubmechanismus sowie die Integration der notwendigen Umfeldsensoren.

A. Mechanische Konstruktion des Fahrwerks

Das Basisgestell des Roboters wurde gezielt asymmetrisch aufgebaut, um die Lastenverteilung zu optimieren. Als Antriebskomponenten kommen zwei servogesteuerte große Motoren zum Einsatz, welche die hintere Achse antreiben. Um die Traktion zu maximieren, wurde im Heckbereich ein Raupenkettantrieb installiert. An der Vorderachse wurden hingegen Räder mit Gummibereifung verwendet. Diese Hybrid-Lösung aus Ketten und Rädern reduziert den Reibungswiderstand

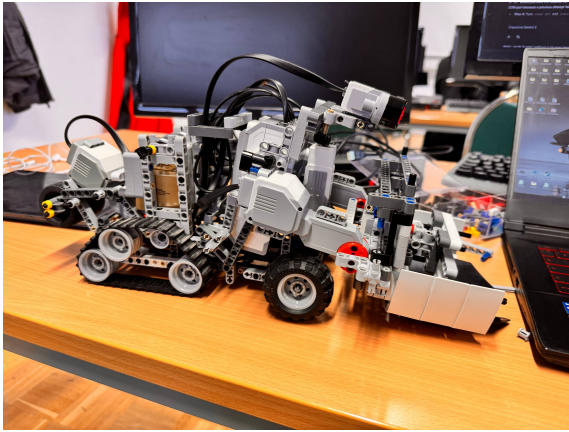


Abbildung 1. Seitenansicht des Robo-Transporters mit sichtbarem Kettenlaufwerk im Heckbereich und dem vorderen Hubwerk.

bei Kurvenfahrten, der bei reinen Kettenfahrzeugen oft zu Ungenauigkeiten in der Odometrie führt.

In Abbildung 1 ist die Anordnung der Antriebskomponenten zu erkennen. Der EV3-Stein, welcher als zentrale Steuereinheit fungiert, wurde mittig platziert, um als Gegengewicht zu den frontal aufgenommenen Lasten zu dienen.

B. Konstruktion des Hubmechanismus

An der Front des Fahrzeugs befindet sich der Gabelstapler-Mechanismus. Dieser wird von einem EV3-Medium-Motor angetrieben, der über eine Zahnradübersetzung mit einer Zahnstange verbunden ist. Durch diese Konstruktion wird die Rotationsbewegung des Motors in eine lineare Vertikalbewegung umgewandelt. Die Gabeln wurden flach und langlaufend konstruiert, um leicht unter Zielobjekte fahren zu können. Die mechanische Belastbarkeit der Zahnstange wurde so gewählt, dass auch bei maximaler Last kein Durchrutschen der Zähne auftritt.

C. Sensorik zur Umfelderkennung

Wie in Abbildung 2 ersichtlich, ist im oberen Frontbereich ein Ultraschallsensor integriert. Dieser ist leicht nach unten geneigt, um kleine Objekte direkt vor der Gabel zu detektieren. Der Sensor arbeitet nach dem Impuls-Echo-Verfahren. Hierbei wird ein Schallsignal emittiert und die Zeit bis zum Eintreffen der Reflexion gemessen. Die Distanz s zum Objekt berechnet sich wie folgt:

$$s = v \cdot \frac{t}{2} \quad (1)$$

In dieser Gleichung repräsentiert v die Ausbreitungsgeschwindigkeit des Schalls in Luft (etwa $343 \frac{\text{m}}{\text{s}}$ bei Raumtemperatur) und t die gemessene Laufzeit des Schallsignals.

IV. SOFTWAREARCHITEKTUR UND STEUERUNG

Die Steuerung des Systems erfolgt aus der Ferne über einen Host-Computer. Hierfür wurde in MATLAB eine objektorientierte Programmstruktur entwickelt.

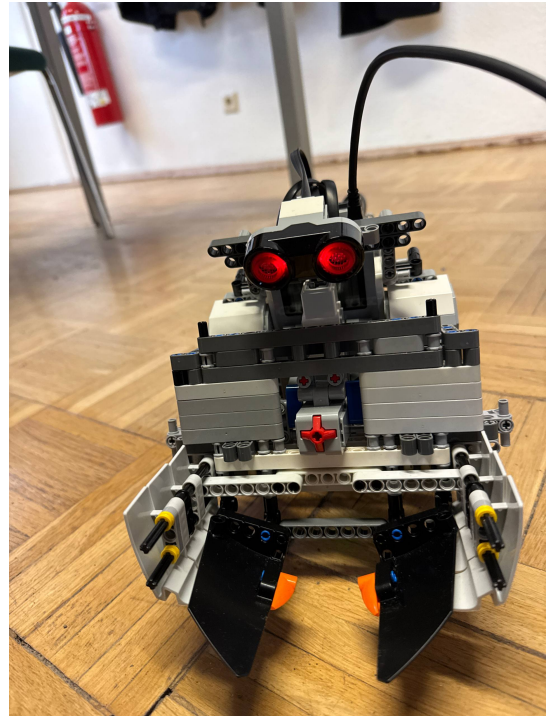


Abbildung 2. Frontansicht des Hubmechanismus und der Sensormontage.

A. Grafische Benutzeroberfläche (GUI)

Über den MATLAB App Designer wurde eine interaktive Steuerungssoftware implementiert. Diese GUI ermöglicht das Herstellen der Bluetooth-Verbindung, das Auslesen von Sensorwerten sowie die manuelle Steuerung der Motoren.

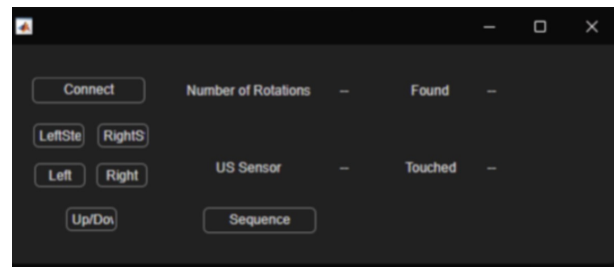


Abbildung 3. Layout der MATLAB-Benutzeroberfläche zur manuellen und teilautonomen Steuerung des Systems.

Wie in Abbildung 3 dargestellt, wurden spezifische Schaltflächen programmiert. Jeder Button löst eine definierte Callback-Funktion aus, welche die entsprechenden Steuerbefehle an den Stein sendet. Die Oberfläche bietet zudem ein Echtzeit-Display für die Distanzwerte des Ultraschallsensors.

B. Ablaufsteuerung des autonomen Modus

Neben der manuellen Steuerung wurde eine Routine für die autonome Objektaufnahme entwickelt. Der Prozess beginnt mit einer Suchrotation. Sobald ein Objekt im Erfassungsbereich des Ultraschallsensors detektiert wird, richtet sich der Roboter aus und fährt geradeaus, bis ein minimaler Schwellwert der Distanz unterschritten wird. Anschließend wird das Fahrzeug gestoppt und der Hebevorgang initiiert. Der vollständige logische Ablauf

dieser Routine ist im Programmablaufplan in Abbildung 4 visualisiert.

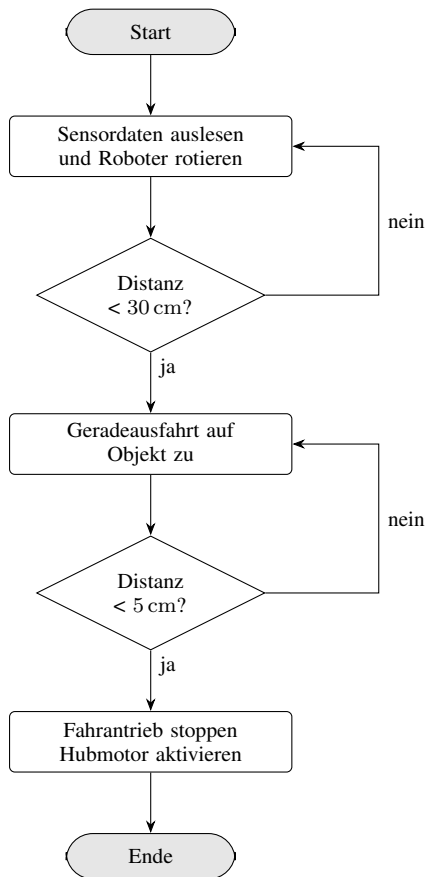


Abbildung 4. Programmablaufplan des teilautonomen Hebealgorithmus.

V. ERGEBNISDISKUSSION

Die Funktionstüchtigkeit des Systems wurde in mehreren Testreihen evaluiert. Dabei wurde besonders auf das Zusammenspiel zwischen mechanischer Lastgrenze und Softwarestabilität geachtet.

A. Mechanische Zuverlässigkeit

Die mechanische Struktur erwies sich als äußerst robust. Der Gabelmechanismus konnte Lasten ohne signifikante Deformationen anheben. Es wurde jedoch beobachtet, dass der Roboter auf glatten Laborböden aufgrund der hohen Traktion der Gummiräder im Frontbereich ein leichtes Ruckeln bei engen Kurvenfahrten aufwies.

Die hinteren Ketten neigten bei plötzlichen Bremsmanövern zu minimalem Schlupf, was in einer leichten Abweichung der Odometriewerte resultierte. Durch die Anpassung der Beschleunigungsrampen in der Software konnte dieser Effekt jedoch weitgehend kompensiert werden. Bei einer Maximalbelastung von über 0,4 kg stieg die Stromaufnahme des Hubmotors stark an, was die thermische Belastung erhöhte.

B. Software- und Verbindungsstabilität

Ein wesentliches Problem während der Entwicklungsphase war die Stabilität der Bluetooth-Kommunikation. In den MATLAB-Konsolenausgaben traten sporadisch Fehler vom Typ `Error using brick.connect` auf. Die Ursache lag in asynchronen Aufrufen innerhalb der Callback-Funktionen, wenn mehrere Steuerbefehle zu schnell hintereinander gesendet wurden. Dies wurde gelöst, indem ein Zustandsautomat (State Machine) implementiert wurde, der sicherstellt, dass Befehle erst gesendet werden, wenn der vorherige Befehl vom Microcontroller quittiert wurde.

VI. ZUSAMMENFASSUNG UND FAZIT

Das Projekt hat gezeigt, dass mit LEGO Mindstorms und MATLAB komplexe Automatisierungsaufgaben erfolgreich modelliert werden können. Es wurde ein funktionsfähiger Roboter-Transporter entwickelt, der Objekte zuverlässig erkennt, anfährt und anhebt. Die Kombination aus manuellem Teleoperationsmodus und autonomer Suchroutine macht das System äußerst flexibel für verschiedene Einsatzszenarien.

Zukünftige Erweiterungen könnten die Integration eines Farbsensors umfassen, um eine sortenreine Logistik zu ermöglichen (z. B. nur grüne Objekte transportieren). Zudem könnte durch den Einsatz eines PID-Reglers für die Radmotoren der Geradeauslauf weiter perfektioniert werden, um auch über größere Distanzen präzise Navigationsergebnisse zu erzielen.

LITERATURVERZEICHNIS

- [1] L. Hering und H. Hering, *Technische Berichte: Gliedern, Gestalten, Vortragen*, 8. Aufl. Springer Vieweg, 2022.
- [2] LEGO Group, *LEGO MINDSTORMS EV3 Benutzerhandbuch*, 2013.
- [3] The MathWorks Inc., *MATLAB App Designer Overview*, 2024.