

Gyroskopisch Gesteuerter Roboter-Arm

„Project MoJo“

Moritz Heucke, Elektrotechnik und Informationstechnik
Otto-von-Guericke-Universität Magdeburg

Zusammenfassung—Das hier dokumentierte Projekt entstand im Zuge des Projektseminars Elektrotechnik/Informationstechnik an der Otto-von-Guericke-Universität Magdeburg. Es handelt sich um einen dreiachsigen Roboter-Arm aus LEGO-Bausteinen, der durch menschliche Eingabe gesteuert wird. Als Eingabemethode dient ein Control-Stick, in dem Gyroskopsensoren verbaut sind. Ziel ist es, dass der Arm stets die räumliche Ausrichtung dieses Control-Sticks ansteuert und so die menschliche Bewegung reproduziert. Diese Arbeit befasst sich mit dem Aufbau und der Funktionsweise sowie Erfolgen und Limitierungen des Projektes.

Schlagwörter—EV3, Gyroskop, LEGO, MATLAB, Roboter

I. EINLEITUNG

E GAL ob in der Forschung, Medizin oder Katastrophenhilfe, von Menschenhand ferngesteuerte Robotik findet zahlreiche Anwendungszwecke. Diese Technologie vermindert Gesundheitsrisiken, kann die Präzision menschlicher Bewegungen verbessern und körperlich belastende Arbeit erleichtern. Im Gegensatz zu autonomen Systemen wird menschliches Urteilsvermögen mit maschineller Kraft und Präzision kombiniert. Auf dieser Idee basiert das hier vorgestellte Projekt. Der Roboter soll durch die gyroskopische Steuerung möglichst intuitiv und natürlich bedient werden können. Dabei tun sich Herausforderungen bei der Konstruktion und Programmierung auf. Denn der Arm soll einen möglichst großen Bewegungsraum haben und die Drehung des Control-Sticks im Raum muss akkurat erfasst werden.

II. VORBETRACHTUNGEN

Im Folgenden werden wichtige Informationen bezüglich genutzter Bauteile gegeben. Ziel ist es, die Funktionsweise des Roboters verständlicher zu machen.

A. Entwicklungsplattform

Als Kontrolleinheit wird der LEGO Mindstorms EV3 verwendet (Abb. 1). Dieser wird mit der Programmiersprache MATLAB in der MATLAB-Entwicklungsumgebung programmiert. Hierzu wird eine von der RWTH Aachen zur Verfügung gestellte Toolbox [1] verwendet, welche die Kommunikation zum EV3 über MATLAB ermöglicht und hilfreiche Kontrollobjekte für Motoren und Sensoren bietet. Auch vereinfacht die Programmiersprache den Umgang mit Vektoren und Matrizen, was für dieses Projekt von Nutzen ist. Der EV3 verfügt im Vergleich zum Vorgängermodell NXT über einen weiteren Motor-Port und eine höhere Rechenleistung [2]. Besonders letzterer Vorteil wird gebraucht, da die Ausführungsgeschwindigkeit des Skripts möglichst hoch sein muss.

B. Gyroskopsensoren

Zur Erfassung von Drehbewegungen werden LEGO-Gyroskope genutzt (Abb. 2). Diese können Rotation jeweils nur um eine Achse messen, weshalb im Control-Stick drei senkrecht zueinanderstehende Sensoren verbaut sind. Dies ermöglicht die Wahrnehmung von Drehungen um alle möglichen im Raum liegenden Achsen. Des Weiteren lassen sich sowohl absolute Winkel als auch Winkelgeschwindigkeiten auslesen. Da zur Umsetzung der Positionserfassung möglichst präzise Winkeländerungen $\Delta\varphi$ gemessen werden müssen, werden in diesem Projekt die Winkelgeschwindigkeiten ω in kleinen Zeitintervallen gemessen und mit der entsprechenden Zeitspanne multipliziert: $\Delta\varphi = \omega \cdot \Delta t$. Dadurch erfolgt eine deutlich genauere Bestimmung der Winkeldifferenzen, weil die absolute Winkelmessung nur ganzzahlige Werte ausgibt. Auch ist zu beachten, dass die Gyroskope beim Einschalten nicht bewegt werden dürfen, da sie sonst falsch kalibriert sind.



Abbildung 1: LEGO Mindstorms EV3 [3]

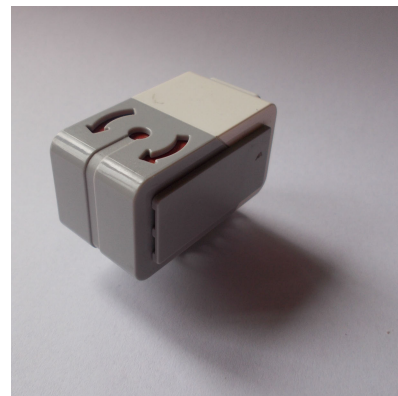


Abbildung 2: LEGO Mindstorms Gyroskop [4]

III. AUFBAU UND FUNKTIONSWEISE

Das Projekt setzt sich aus Hardware- und Softwarekomponenten zusammen. In diesem Abschnitt wird auf die bauliche und programmiertechnische Umsetzung eingegangen.

A. Konstruktion

Die baulichen Hauptmerkmale umfassen den Control-Stick, den LEGO EV3, den Roboter-Arm mit drei Motoren sowie einen Tastsensor als Stopp-Knopf, der am unteren Gerüst des Roboters befestigt ist (Abb. 3, 4). Die Steuerung erfolgt über den Control-Stick, in dessen unterem Teil drei Gyroskope verbaut und über LEGO-Datenkabel mit dem EV3 verbunden sind. Der obere Teil dient als Griff und ein Standfuß erleichtert die Kalibrierung der Sensoren. Die von den Gyroskopen erfassten Winkelgeschwindigkeiten werden vom EV3 ausgelesen und durch ein MATLAB-Skript verarbeitet.

Der Arm besteht aus einem rotierenden Basissegment, auf dem ein weiteres Segment angebracht ist, das um eine orthogonale Achse rotiert. Ein zusätzliches Gelenk verbindet das mittlere Segment mit einer Plattform, auf der Sensoren oder andere Instrumente installiert werden können. Die Rotationen der unteren beiden Gelenke bewirken eine Ausrichtung des Arms in die Richtung, in die der Control-Stick zeigt. Das obere Gelenk ermöglicht eine zusätzliche Drehung der Plattform um ihre eigene Achse, wodurch die exakte Orientierung des Control-Sticks im Raum nachgebildet wird. Die gesamte Konstruktion ist auf einem stabilen Gerüst montiert, das für einen sicheren Stand sorgt. An Arm und Stick ist je ein blauer Pin angebracht, mit dem die richtige Startorientierung sichergestellt werden kann.

B. MATLAB-Skript

Nach der Initialisierung der Motor- und Sensorobjekte sowie der Definition globaler Variablen wird eine While-Schleife ausgeführt, die erst durch das Betätigen des Stopp-Knopfs beendet wird. Innerhalb dieser Schleife werden kontinuierlich Sensordaten erfasst und daraus die Soll-Winkel für die Motoren berechnet. Anschließend erfolgt die Ansteuerung der Motoren gemäß den berechneten Werten. Beim Drücken des Stopp-Knopfs kehren die Motoren in ihre Ausgangsposition zurück. Abbildung 5 zeigt eine schematische Darstellung des Skripts.

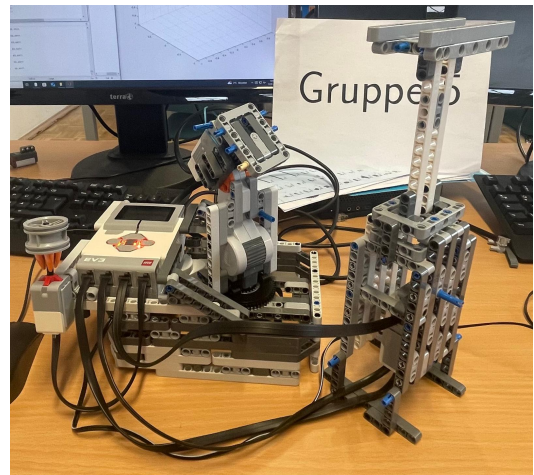


Abbildung 4: Finale Umsetzung des Roboters

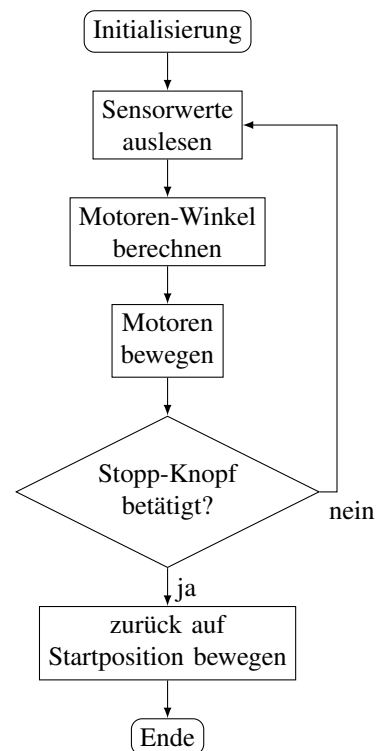


Abbildung 5: Programmablaufplan des MATLAB-Skripts

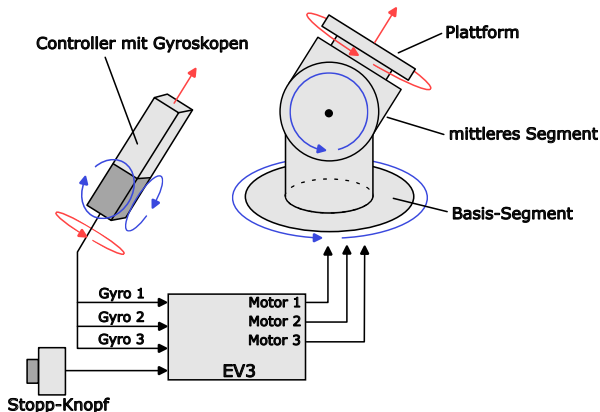


Abbildung 3: Aufbau des Roboters

C. Erfassung der Ausrichtung des Control-Sticks

Eine zentrale Herausforderung in der Entwicklung besteht darin, dass die Gyrodaten nicht direkt zur Steuerung der Motoren verwendet werden können. Der Grund hierfür liegt in der sich ständig verändernden Orientierung der Rotationsachsen der Gyroskope während der Bewegung des Control-Sticks. Jede Gyroskopachse kann frei im Raum gedreht werden. Im Gegensatz dazu kann das Basissegment des Arms nur um eine räumlich feste Achse rotieren, während die nachfolgenden Segmente jeweils um eine weitere bewegliche Achse rotieren. Lediglich das obere Plattformsegment kann direkt über den Gyroskopwert des entsprechenden Sensors gesteuert werden, da beide Drehachsen parallel zueinander ausgerichtet sind. Dieser

Zusammenhang ist in Abbildung 3 farblich dargestellt. Zur Lösung dieses Problems wird ein System aus drei orthogonalen Einheitsvektoren aufgestellt, welche die Gyroskopachsen repräsentieren. Einer dieser Vektoren dient als Richtungsvektor, an dem sich der Roboter-Arm ausrichtet. In den Abbildungen 3 und 6 ist dieser rot gefärbt. Während der Bewegung des Control-Sticks müssen die Vektoren so transformiert werden, dass sie stets die Ausrichtung der Gyrosachsen übernehmen. Die Basisvektoren werden in zwei Matrizen $[\mathbf{V}]$, $[[\mathbf{V}_t]] \in \mathbb{R}^{3 \times 3}$ als Spaltenvektoren gespeichert. Die Matrix $[\mathbf{V}]$ enthält die Ausgangsvektoren vor der Transformation und $[\mathbf{V}_t]$ enthält die transformierten Vektoren. Vor der Rotation sind $[\mathbf{V}]$ und $[\mathbf{V}_t]$ identisch. Nach der Messung der drei Gyroskopwinkel wird jeder der Spaltenvektoren in $[\mathbf{V}_t]$ nacheinander um alle Spaltenvektoren in $[\mathbf{V}]$ um die gemessenen Winkel φ rotiert. Hierfür eignet sich die „Rodrigues’ rotation formula“ (1), mit der ein beliebiger Raumvektor um eine definierte Achse und einen bestimmten Winkel rotiert werden kann [5].

$$\vec{v}_{\text{rot}} = \vec{v} \cos \varphi + (\vec{k} \times \vec{v}) \sin \varphi + \vec{k}(\vec{k} \cdot \vec{v})(1 - \cos \varphi) \quad (1)$$

\vec{v}_{rot}	rotierter Vektor
\vec{v}	zu rotierender Vektor
\vec{k}	Rotationsachse (Einheitsvektor)

Sowohl \vec{v} als auch \vec{k} sind Spaltenvektoren von $[\mathbf{V}]$. Die Berechnung der Gesamtrotation eines Vektors erfolgt durch die aufeinanderfolgende Anwendung von Teilrotationen. Diese Teilrotationen sind die von den Gyroskopen gemessenen Winkeldifferenzen $\varphi_{x,y,z}$. Diese Herangehensweise basiert auf der kommutativen Additivität von Winkelgeschwindigkeiten. Das Prinzip wird in Quelle [6] näher erläutert.

$$\begin{aligned} \vec{\omega}_{\text{ges}} &= \vec{\omega}_x + \vec{\omega}_y + \vec{\omega}_z \\ \vec{e}_r \frac{d\varphi_{\text{ges}}}{dt} &= \vec{e}_x \frac{d\varphi_x}{dt} + \vec{e}_y \frac{d\varphi_y}{dt} + \vec{e}_z \frac{d\varphi_z}{dt} \\ \vec{e}_r d\varphi_{\text{ges}} &= \vec{e}_x d\varphi_x + \vec{e}_y d\varphi_y + \vec{e}_z d\varphi_z \end{aligned} \quad (2)$$

$\vec{\omega}_{\text{ges}}$	Gesamtwinkelgeschwindigkeit
$\vec{\omega}_{x,y,z}$	Teilgeschwindigkeiten um Gyro-Achsen
\vec{e}_r	Einheitsvektor parallel zur Rotationsachse
$\vec{e}_{x,y,z}$	Einheitsvektoren parallel zu Gyro-Achsen

Da die Addition von Teildrehungen zu einer Gesamtdrehung nur für differenzielle Winkel exakt ist, müssen möglichst kleine Winkelveränderungen $\Delta\varphi$ erfasst werden, um eine präzise Steuerung des Roboter-Arms zu gewährleisten. Nach der Transformation werden die Spaltenvektoren aus $[\mathbf{V}_t]$ in $[\mathbf{V}]$ übernommen und bilden die Rotationsachsen für die nächste Transformation. Der Prozess ist schematisch in Abbildung 6 dargestellt. Zuletzt wird der zuvor festgelegte Richtungsvektor in Kugelkoordinaten (r, α, β) umgewandelt. Die Winkel α und β können nämlich direkt von den Motoren angesteuert werden, damit der Arm die Ausrichtung des Vektors übernimmt.

D. Steuerung der Motoren

Zum Antrieb des Roboters werden Servomotoren genutzt, was bedeutet, dass ihre aktuelle Position jederzeit ausgelesen werden kann. Außerdem ist die Eigenschaft „speedRegulation“ aktiviert, wodurch die Geschwindigkeit direkt durch die Eigenschaft „power“ kontrolliert werden kann [1]. Die Bewegung eines Motors auf eine spezielle Position wurde folgendermaßen umgesetzt. Der jeweilige Motor wird so lange in die entsprechende Richtung bewegt, bis die angestrebte Position erreicht ist. Dann wird er gestoppt. Unter diesen Voraussetzungen ist festzustellen, dass die LEGO-Motoren, gerade bei höheren Drehzahlen, nicht an der exakten Position zum Stehen kommen. In Folge wird die Zielposition leicht übertreten.

Dieses Problem löst eine Funktion im Code, welche die Drehzahl und Drehrichtung eines Motors abhängig von der Differenz zwischen der angestrebten und aktuellen Position bestimmt. Bei Annäherung an die angestrebte Position wird dabei die Geschwindigkeit verringert. Die Funktion nimmt verschiedene Parameter auf, die das Verhalten der Motoren beeinflussen. Hierzu zählen der anzusteuende Winkel und der Toleranzbereich, also die Distanz, ab der die Drehzahl verringert wird. Auch die Normalgeschwindigkeit außerhalb dieses Bereiches und die Minimalgeschwindigkeit, auf die der Motor verlangsamt wird, wenn die Positionsdivergenz gegen Null geht. Aus verschiedenen Testläufen stellen sich die folgenden Parameterwerte als geeignet heraus.

Toleranz:	25 Grad
Maximalgeschwindigkeit („power“):	5
Minimalgeschwindigkeit („power“):	0

Die Geschwindigkeitskurve unter diesen Parametern ist in Abbildung 7 festgehalten. Bei diesem Ansatz zeigt sich in der Praxis aber auch eine nennenswerte Auffälligkeit. Aufgrund der geringen Minimal- und Maximalgeschwindigkeit ist die berechnete Geschwindigkeit in einem Differenzbereich von mehreren Grad zu gering, um von den LEGO-Motoren umgesetzt zu werden. Dies hat zur Folge, dass die Motoren innerhalb dieses Bereiches stillstehen und sich nach Verlassen recht ruckartig in Bewegung setzen. Dies verringert die Genauigkeit und Reaktivität des Arms, aber filtert gleichzeitig kleine, ungewollte Bewegungen heraus.

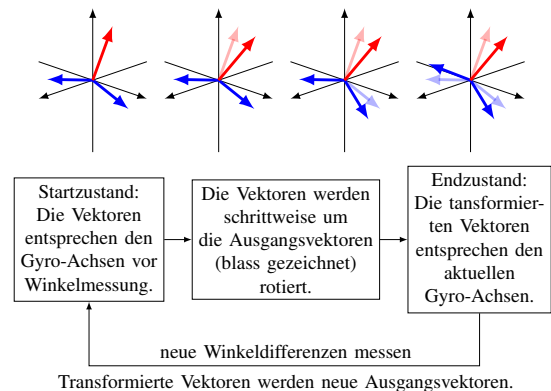


Abbildung 6: exemplarischer Ablauf der Vektortransformation

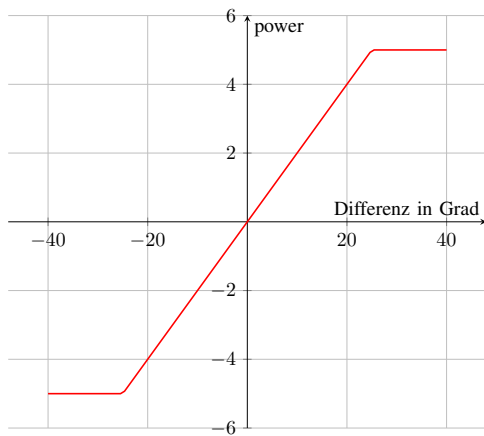


Abbildung 7: Motorverhalten abhängig von der Nähe zur Zielposition

IV. ERGEBNISDISKUSSION

Die Umsetzung des Konzeptes ist gelungen. Der im Rahmen des Projektseminars entwickelte Roboter erfüllt seine vorgesehene Funktion, die Orientierung des Control-Sticks im Raum anzusteuern. Aber es zeigen sich funktionale Limitierungen, die durch die LEGO-Bauteile bedingt sind. Ein zentrales Problem besteht in der Genauigkeit der Positionserfassung des Control-Sticks. Die Ausführungsgeschwindigkeit des MATLAB-Skripts auf dem EV3 ist zu gering, als dass eine exakte Positionserfassung über Zeiträume mehrerer Minuten möglich ist. Sind die gemessenen Winkeldifferenzen größer, so ist die Transformation der Vektoren ungenauer. Bei zu großen Zeitabständen zwischen den Messungen und zu schnellen Bewegungen des Control-Sticks, kann dieser Effekt bereits nach Sekunden spürbar werden. Daher sind langsame und gleichmäßige Bewegungen des Control-Sticks notwendig. Weitere Problemstellen sind die Latenz, mit welcher der Roboter auf Eingaben reagiert und Einschränkungen in der Beweglichkeit durch die kurzen und unflexiblen Kabel. Durch Optimierungen in der Konstruktion und Programmierung funktioniert der Roboter dennoch in ausreichendem Maße. Das Ergebnis zeigt, dass das Konzept intuitiv bedienbarer Robotik umsetzbar ist und bietet einen Anhaltspunkt für weitere Entwicklung.

V. ZUSAMMENFASSUNG UND FAZIT

Der Roboter-Arm verfügt über 3 Bewegungsachsen und ist so in der Lage jegliche Orientierung des Control-Sticks nachzustellen. Die Steuerung erfolgt über einen LEGO EV3, der ein MATLAB-Skript ausführt. Hierbei werden Sensordaten eingelesen, verarbeitet und schließlich in Bewegungsanweisungen umgesetzt. Eine eindeutige Erfassung der Positionierung wird durch Transformation von Vektoren erzielt. Die Motoren werden durch eine Kontrollfunktion gesteuert, welche die Zielgenauigkeit erhöht. Auch wenn die LEGO-Bauteile Begrenzungen in der Leistungsfähigkeit mit sich bringen, erfüllt der Roboter seine Anforderungen und stellt eine Möglichkeit menschlich gesteuerter Robotik dar. Weitere Entwicklungsschritte können die Verwendung einer leistungstärkeren Steuereinheit und eine kabellose Datenübertragung der Sensordaten umfassen. Der Roboter ist als eine Basis konzipiert, die durch verschiedenste Instrumente erweitert werden kann. Denkbar sind etwa eine Kamera und andere Sensorik oder eine Greifzange. Weitergedacht kann diese Technologie Menschen helfen, ihnen unzugängliche Räume zu erkunden, wie die Tiefsee oder Objekte im Weltraum. Sie ermöglicht es, Handlungen in großen Entfernungen auszuführen, die Erfahrung und Urteilsvermögen eines Menschen erfordern. Somit bietet das hier vorgestellte Projekt nicht nur eine technische Lösung, sondern öffnet auch Perspektiven für weiterführende Anwendungen an der Schnittstelle zwischen Mensch und Roboter.

LITERATURVERZEICHNIS

- [1] ATORF, L. ; SONDERMANN, B. ; STADTMANN, T. ; ROSSMANN, J.: *RWTH - Mindstorms EV3 Toolbox*. <https://git.rwth-aachen.de/mindstorms/ev3-toolbox-matlab>. Version: 2018
- [2] WIKIPEDIA, THE FREE ENCYCLOPEDIA: *Lego Mindstorms EV3*. https://en.wikipedia.org/wiki/Lego_Mindstorms_EV3. Version: Februar 2025
- [3] VETALRU: *Lego mindstorms ev3*, Oktober 2013. <https://commons.wikimedia.org/w/index.php?curid=28877049>. – CC BY-SA 3.0, Bild unverändert, Abruf: 26.02.2025
- [4] JIM MCTURBO: *EV3 Gyro Sensor*, Juni 2020. <https://commons.wikimedia.org/w/index.php?curid=91295885>. – CC BY-SA 4.0, Bild unverändert, Abruf: 26.02.2025
- [5] WIKIPEDIA, THE FREE ENCYCLOPEDIA: *Rodrigues' rotation formula*. https://en.wikipedia.org/wiki/Rodrigues%27_rotation_formula. Version: Februar 2025
- [6] WIKIPEDIA, THE FREE ENCYCLOPEDIA: *Winkelgeschwindigkeit*. <https://de.wikipedia.org/wiki/Winkelgeschwindigkeit>. Version: Februar 2025