

Light-Painting-Robot 2.0

Tobias Wagner, Elektromobilität
 Otto-von-Guericke-Universität Magdeburg

Abstract— Im Rahmen des jährlich stattfindenden Projektseminars Elektrotechnik/Informationstechnik wurde die Idee eines Light-Painting-Roboters weiterentwickelt. Entstanden ist der Light-Painting-Robot 2.0, der mit Hilfe eines LEGO-Mindstorms-Sets und einer MATLAB-Programmierung umgesetzt wurde. Mit dem Roboter können verschiedene geometrische Formen sowie einfache selbstgezeichnete Bilder realisiert werden.

Schlagwörter—Calliope mini, LEGO Mindstorms, Light-Painting, MATLAB, Roboter

I. EINLEITUNG

LIGHT-PAINTING [1] bezeichnet eine Technik aus der Fotografie. Mittels Langzeitbelichtung können Formen oder Linien erzeugt werden, indem eine Lichtquelle vor der Kamera bewegt wird. Die besten Ergebnisse werden bei Dunkelheit erzielt. Das Gezeichnete wird erst im fertigen Bild sichtbar. Dadurch ist es für den Künstler schwierig zu erkennen, wo er bereits gezeichnet hat. Möchte man eine geschlossene Form zeichnen, ist es nur mit Glück und vielen Versuchen möglich, dass sich Anfangs- und Endpunkt der Zeichnung in einem Punkt treffen. Auch die Belichtungszeit spielt eine wichtige Rolle. Je länger diese ist, desto mehr Rauschen ist im Bild zu sehen [2]. Der Light-Painting-Robot 2.0 soll diese Probleme lösen. Mit ihm ist es möglich, möglichst schnell ein präzises Bild mit klaren Linien zu zeichnen, um die Belichtungszeit zu minimieren.

II. VORBETRACHTUNGEN

Zur Realisierung des Roboters wurde ein bereits bestehendes mechanisches Konzept verwendet. Zusätzlich zu den Teilen, die in dem LEGO Bausatz enthalten sind, wurden noch einige weitere Materialien verwendet.

A. Inspiration

Die ursprüngliche Idee eines Light-Painting-Roboters stammt aus dem Projekt „Sally can Draw: Automatisierter Light-Painting Roboter“ [3] aus dem Jahr 2022. Nach mehreren Konstruktionsansätzen fehlte im letzten Jahr die Zeit, um das Endergebnis zu perfektionieren. An diesem Punkt setzt der Light-Painting-Robot 2.0 an, wobei die mechanische Grundstruktur des Roboters erhalten bleibt. Eine schrittweise Optimierung der Mechanik und eine neue Software sind die wesentlichen Unterschiede.

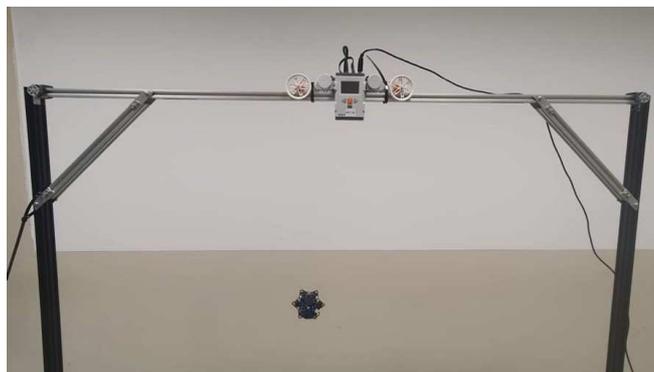
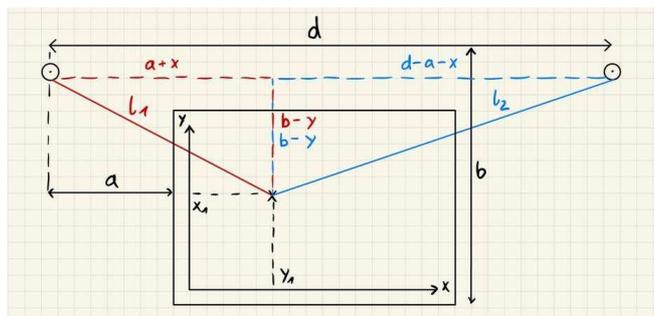


Abbildung 1: Aufbau des Light-Painting-Robot 2.0

B. Geometrie

Um zeichnen zu können, benötigt der Roboter zunächst einen vorgegebenen Rahmen. Dieser Rahmen ist in diesem Fall ein Koordinatensystem, da das MATLAB-Programm dieses benötigt, um die Größe und Position des zu zeichnenden Bildes bestimmen zu können. Um die LED wie gewünscht bewegen zu können, müssen die Schnüre entsprechend auf- oder abgerollt werden. Für jeden Punkt im Koordinatensystem ergibt sich eine bestimmte Länge für die linke und eine für die rechte Schnur. Als Bezugspunkte für die Längen dienen die Umlenkrollen, die am Rand des Gestells angebracht sind. Mit Hilfe des Satzes des Pythagoras können die erforderlichen Schnurlängen für jede x/y-Kombination berechnet werden. Abbildung 2 zeigt die zugrunde liegende Berechnung. Bestimmt man die Längendifferenz der Schnüre zwischen zwei Punkten, so kann die LED von einem Punkt zum anderen bewegt werden. Dieses Prinzip bildet die Grundlage für das Zeichnen verschiedener Formen.



$$l_1 = \sqrt{(a+x)^2 + (b-y)^2} \quad l_2 = \sqrt{((d-a-x)^2 + (b-y)^2)}$$

Abbildung 2: Geometrie zur Berechnung der Schnurlängen

C. Material

Da der Roboter nicht allein mit LEGO Mindstorms realisierbar ist, werden Zusatzkomponenten benötigt. Die größte Komponente ist das Metallgestell, auf dem die LEGO-Komponenten montiert werden. Als Lichtquelle wird ein Calliope mini [4] verwendet. Dieser Mikrocontroller bietet viele Funktionen, wobei in diesem Projekt der Schwerpunkt auf der Konstruktion und Programmierung des Roboters liegt und daher nur die RGB-LED des Calliope mini verwendet wird. Die Lichtquelle wird mit Hilfe von zwei Angelschnüren aufgehängt. Diese sind sehr dünn und reißfest und daher optimal geeignet. Die Stromversorgung des Calliope mini erfolgt über einen Akku, der auf der Rückseite angebracht ist.

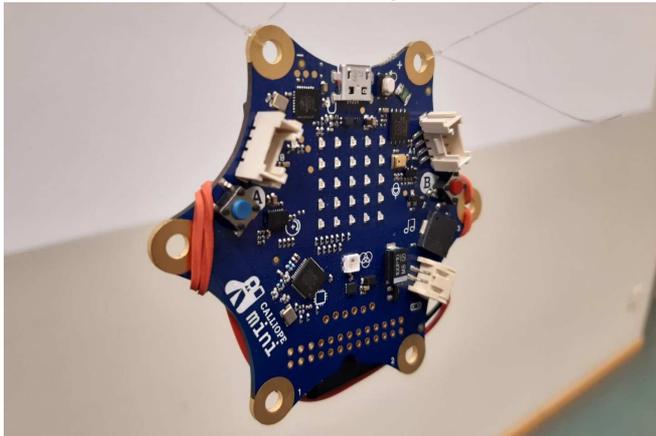


Abbildung 3: Calliope mini an Schnüren befestigt

III. ENTWICKLUNGSPROZESS

Im Bereich der mechanischen Konstruktion konnten schrittweise Verbesserungen erzielt werden, und im Bereich der Programmierung konnte nach mehreren Ansätzen eine Lösung gefunden werden.

A. Konstruktion

Ausgehend von der letztjährigen Konstruktion wurde mit der Entwicklung begonnen. Durch die Platzierung der Motoren auf dem NXT-Stein wurde das Problem der begrenzten Kabellänge zur Verbindung der Komponenten umgangen. An jedem Motor ist ein Rad befestigt, auf dem die Schnur aufgewickelt wird. Durch Umlenkrollen am äußeren Rand des Gestells ist es möglich, fast die gesamte Breite des als Zeichenfläche zu nutzen. Eine drehbare Lagerung der Umlenkrollen minimiert die Reibung der Schnur und reduziert somit Vibrationen. Die Inkompatibilität der LEGO Elemente mit dem Metallrahmen stellt ein Problem bei der Befestigung dar. Im Entwicklungsprozess wurden die ursprünglich verwendeten Kabelbinder durch eine Haltekonstruktion aus LEGO ersetzt. Diese Konstruktion sowie die Umlenkrollen werden mit Gummibändern fixiert, um Bewegungen der Bauteile auf dem Metallrahmen zu verhindern. Auf der Rückseite des NXT-Steins befinden sich vier Schalter, mit denen die LED manuell ausgerichtet werden kann. So kann im Voraus festgelegt werden, wo der Roboter mit dem Zeichnen des Bildes beginnen soll.



Abbildung 4: NXT und Motoren

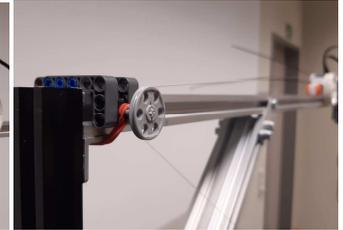


Abbildung 5: Umlenkrolle

B. Programm

Bei der Erstellung des MATLAB-Codes wurden zwei Ansätze verfolgt. Diese unterscheiden sich im Wesentlichen in der Art der Motoransteuerung.

1) *Motoransteuerung durch Tacholimit:* Mit dieser Methode war es nicht möglich präzise Linien abzufahren. Beim ersten Ansatz werden die Motoren mit einer konstanten Leistung und einer vorher berechneten Tachobegrenzung für die zu fahrende Strecke versehen. Die Motoren müssen also eine bestimmte Anzahl von Umdrehungen bei konstanter Drehzahl ausführen und dann anhalten. Bei diesem Ansatz erreichen die Motoren ihr Tacholimit jedoch zu unterschiedlichen Zeitpunkten. Mit dieser Methode war es nicht möglich, präzise Linien abzufahren.

2) *Motoransteuerung durch Power:* Bei dieser Variante werden die Motoren in kurzen Zeitabständen mit einer jeweils neu berechneten Motorpower beaufschlagt. Dazu muss zunächst eine Motorkalibrierung durchgeführt werden. Dabei wird die Leistung in Fünferschritten von 0 auf 100 % erhöht. Proportional dazu wird das Tacholimit erhöht. Theoretisch sollte die Zeit bis zum Erreichen des Tacholimits konstant sein. In der Praxis war dies jedoch nicht der Fall. Als Grund für die aufgetretenen Ungenauigkeiten kann die unzureichende Datenübertragungsgeschwindigkeit mit dem in die Jahre gekommenen NXT-Stein genannt werden. Trotzdem sind die Ergebnisse bis zu einer Leistung von 80% ausreichend genau, weshalb diese als maximale Power definiert wurde. Der größte Teil der Programmierung besteht aus Berechnungen, die dem eigentlichen Zeichnen vorausgehen. Zuerst wird die Form, die gezeichnet werden soll, in MATLAB gezeichnet. Diese Form wird in n Teile aufgeteilt, so dass viele kleine Abschnitte entstehen. Später kann für jeden zu zeichnenden Abschnitt eine spezifische Power an die beiden Motoren gesendet werden. Zuvor wird die erforderliche Längenänderung der Schnüre und die daraus resultierende Winkeländerung am Motor berechnet. Die resultierende Power wird anschließend noch geglättet, um einen weicheren Übergang zwischen den Ansteuerungen zu erreichen. Zusätzlich ist die Zeit, die der Roboter zum Zeichnen des Bildes benötigt, variabel einstellbar. Dies ermöglicht eine einfache Anpassung an die gewünschte Belichtungszeit, mit der das Bild aufgenommen werden soll. Auf diese Weise können geometrische Formen gezeichnet werden. Um ein eigenes Bild vom Roboter zeichnen zu lassen, wird auf ein externes MATLAB-Programm [5] zurückgegriffen, welches die eigene Zeichnung einliest und so anpasst, dass sie vom Roboter gezeichnet werden kann.

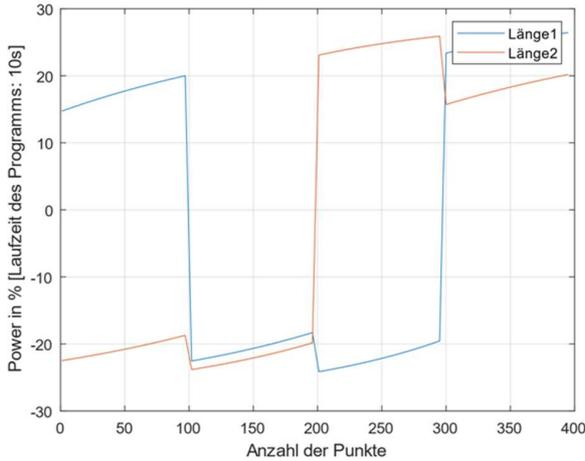


Abbildung 6: Motorpower beim Zeichnen eines Rechtecks

IV. ERGEBNISDISKUSSION

Das Ergebnis ist angesichts der begrenzten Zeit von zwei Wochen sehr zufriedenstellend. Der Roboter ist in der Lage, die vorgegebenen Formen sehr genau und gut reproduzierbar zu zeichnen. Die Probleme mit dem Wackeln des Calliope mini wurden durch mehrere Maßnahmen minimiert. Zum einen durch den auf der Rückseite angebrachten Akku. Durch das erhöhte Gewicht ist die Lichtquelle weniger anfällig für auftretende Vibrationen. Auch die Glättung der Übergänge bei Richtungsänderungen sowie die drehbar gelagerten Umlenkrollen erzielen diesen Effekt. Das Ergebnis des Freihandzeichnens ist noch verbesserungswürdig. Das Zeichnen von Schriftzügen ist schwierig, da man schon bei wenigen Buchstaben an die Grenze der maximal sinnvollen Belichtungszeit stößt. Außerdem muss eine sehr hohe Power gewählt werden, was zu einem stärkeren Wackeln der Lichtquelle und damit zu ungenauem Zeichnen führt. Nachdem die endgültige Programmiermethode festgelegt war, traten keine größeren Probleme mehr auf. Die Aufgabe bestand fortan darin, die Funktion zu optimieren und kleinere Fehler zu beheben. Die Verbesserung der Benutzerfreundlichkeit wurde schließlich durch die Erstellung von Funktionen in MATLAB und das Hinzufügen der manuellen Steuerung erreicht.



Abbildung 7: Ergebnisbild Stern

V. ZUSAMMENFASSUNG UND FAZIT

Zusammenfassend kann gesagt werden, dass der Light-Painting-Robot 2.0 seinen Zweck gut erfüllt. Er stellt eine Weiterentwicklung gegenüber dem letztjährigen Projekt dar und bietet zudem mehr Funktionen. Die Zeichenfläche ist größer und die gesamte Konstruktion ist stabil. Verschiedene Formen können gezeichnet werden. Erst bei komplexeren Zeichnungen stößt der Roboter an seine Grenzen. Hier gibt es den größten Verbesserungsbedarf. Auch das Ergebnisbild des Sterns (Abbildung 7) kann sicherlich noch perfekter gezeichnet werden. Mit einigen Verbesserungen in der Programmierung oder der Konstruktion sollte hier ein nahezu perfektes Bild im Bereich des Möglichen liegen. Durch eine Implementierung des Calliope mini wäre es möglich, die Farben während des Zeichnens zu ändern oder die LED zu bestimmten Zeiten ein- und auszuschalten. Damit wäre es auch möglich, Bilder zu erzeugen, die nicht aus einer geschlossenen Linie bestehen. Im Großen und Ganzen funktionieren die Grundfunktionen des Light-Painting-Robot 2.0 sehr gut und können mit mehr Zeit noch weiter ausgebaut und perfektioniert werden.

LITERATURVERZEICHNIS

- [1] WIKIPEDIA: *Light Painting*. https://de.wikipedia.org/wiki/Light_Painting. Stand: Februar 2023.
- [2] Cyber-shot Benutzeranleitung: *Bulb*. <https://helpguide.sony.net/dsc/1210/v1/de/contents/02/03/07/07.html>. Stand: Februar 2023
- [3] LEGO-Praktikum. Entwickeln, programmieren, optimieren: *Gruppe 3*. <https://journals.ub.ovgu.de/index.php/LEGO/article/view/2072/2065>. Stand Februar 2023.
- [4] Calliope: *Calliope mini Übersicht*. <https://calliope.cc/calliope-mini/uebersicht> Stand: Februar 2023.
- [5] MathWorks: *How can I draw with mouse as we do in paint? :* <https://de.mathworks.com/matlabcentral/answers/320227-how-can-i-draw-with-mouse-as-we-do-in-paint> Stand: Februar 2023.