

# 3D-Puzzle mit Greifarm

Maja Wantke `mwantke@stud.hs-bremen.de`  
Lara Miritz `lmiritz@stud.hs-bremen.de`  
Nikias Scharnke `nscharnke@stud.hs-bremen.de`  
Sara-Ann Wong `swong@stud.hs-bremen.de`  
Angewandte Mathematik für Medieninformatik  
Hochschule Bremen

July 25, 2024

## Abstract

Diese Dokumentation beschreibt die Entwicklung und Implementierung einer 2D Roboterarm Simulation, die Konzepte der Kinematik praxisnah veranschaulicht. Die Simulation ermöglicht es, einen Roboterarm interaktiv zu steuern und Puzzleteile zu bewegen. Nutzer können den Roboterarm selbst konfigurieren und so die Komplexität der Steuerung kennenlernen.

## 1 Motivation

Roboterarmsimulationen sind ein wesentlicher Bestandteil der Robotikforschung. Sie ermöglichen es, komplexe Bewegungsabläufe zu visualisieren und zu verstehen. Unser Projekt soll eine benutzerfreundliche und interaktive Anwendung bereitstellen, die es dem Benutzer ermöglicht, die Funktionsweise eines Roboterarms durch direkte Interaktion zu erforschen. Der Benutzer kann dabei selbst experimentieren und forschen, um eigene Erkenntnisse über die Bewegungssteuerung und Kinematik des Roboterarms zu gewinnen. Dies fördert das Verständnis und die Fähigkeit, theoretisches Wissen in die Praxis umzusetzen.

## 2 Verwandte Arbeiten

Unser Projekt wurde durch verschiedene Arbeiten im Bereich der Roboterarmsimulation inspiriert. Die Bachelorarbeit zur Steuerung eines 5-DOF Handhabungsroboters in Arbeitsraumkoordinaten“[1] von Julia Dubcova erklärt die Grundlagen der Robotersteuerung und der inversen Kinematik, die wir für unseren Roboterarm nutzen. Eine weitere wichtige Quelle war die Veröffentlichung Tactile Robotic Assembly“[2] vom BBSR. Diese half uns, die Interaktion des Roboterarms mit Objekten zu verstehen. Beide Arbeiten haben uns wertvolle Ideen und Techniken geliefert, die wir in unserer interaktiven Roboterarmsimulation anwenden wollen. Ergänzend haben wir uns auch an der Arbeit Robot Arm Control using Machine Learning“[3] orientiert, welche moderne Ansätze zur Steuerung von Roboterarmen mittels maschinellem Lernen beleuchtet. Zur Umsetzung der Animationen orientieren wir uns an der neuesten Auflage des Werkes Computer Animation: Algorithms and Techniques“[4] von Rick Parent, die unter anderem die Grundlagen der Animationsprogrammierung vermittelt. Es behandelt Grundlagen wie Bewegung, Deformation und physikalische Simulation, um realistische Animationen zu erstellen.

## 3 Projektübersicht

In diesem Projekt wurde ein 2D-Puzzle-Spiel entwickelt, das durch einen Roboter-Greifarm gelöst werden muss. Der Greifarm wird durch Drag Drop interaktiv gesteuert und verwendet inverse Kinematik, um Bewegungen präzise auszuführen. Diese Steuerung ermöglicht es dem Benutzer, die Komplexität der Roboterbewegungen zu verstehen und verschiedene Konfigurationen des Arms auszuprobieren.

Der Benutzer kann Ziele per Mausinteraktion festlegen, die der Roboterarm erreichen soll, und Puzzleteile greifen, um sie an eine Position im Raster zu verschieben. Dabei wird eine Kollisionserkennung verwendet, die überprüft, ob eine Kollision zwischen einem Puzzleteil und dem Raster auftritt. Ist dies der Fall, so rastet das Puzzleteil im Raster ein. Durch diese interaktive und benutzerfreundliche Anwendung erhält der Benutzer die Möglichkeit, die Funktionsweise eines Roboterarms praktisch zu erforschen und dabei eigene Experimente durchzuführen. Dies fördert ein tieferes Verständnis der Bewegungsabläufe und der Kinematik von Roboterarmen und ermöglicht wertvolle

Erkenntnisse durch direkte Interaktion.

## 4 Entwicklungsprozess

Der Entwicklungsprozess begann mit der Ideenfindung und Planung, bei der verschiedene Konzepte für die Entwicklung eines Spiels, das von einem Roboterarm gelöst werden sollte, erörtert wurden. Das Ziel war es, eine benutzerfreundliche Anwendung zu schaffen, die es Nutzern ermöglicht, die Funktionsweise eines Roboterarms durch direkte Interaktion zu erforschen.

Am 10. Juni 2024 wurde das Exposé abgegeben. Zu diesem Zeitpunkt wurde eine Grundstruktur der Anwendung erstellt, die das Layout und die grundlegenden Funktionen umfasste. Diese Struktur bildete die Grundlage für die weitere Entwicklung.

Die erste Besprechung, die am 26. Juni 2024 stattfand, ermöglichte eine umfassende Überprüfung des bisherigen Fortschritts. Dabei wurde entschieden, die Konfiguration der Anwendung zu erweitern, um den Nutzern eine detailliertere Untersuchung der Kinematik des Roboterarms zu ermöglichen. Diese Erweiterung war nicht von Anfang an geplant, sondern entstand aus der Erkenntnis heraus, den Benutzern mehr Flexibilität bei der Erkundung der Armkinematik zu bieten.

Am 12. Juli 2024 wurde der aktuelle Stand der Entwicklung vorgestellt. In dieser Präsentation lag der Schwerpunkt auf der neuen Konfiguration, die es den Nutzern ermöglicht, die Kinematik des Roboterarms selbst zu erforschen. Die Demonstration zeigte, wie Benutzer verschiedene Parameter anpassen und die Auswirkungen dieser Anpassungen in Echtzeit beobachten konnten. Diese Funktion ermöglichte eine tiefere Interaktion mit der Anwendung und stellte einen wesentlichen Fortschritt im Vergleich zur ursprünglichen Planung dar.

Vor der Präsentation wurden umfassende Tests durchgeführt, um die Funktionsfähigkeit der Implementierung sicherzustellen. Dazu gehörten Unittests zur Überprüfung der verschiedenen Komponenten der Anwendung, einschließlich der Kollisionserkennung und der Drag-and-Drop-Funktionalität. Diese Tests stellten sicher, dass alle Funktionen stabil und fehlerfrei arbeiteten.

Nach der Präsentation am 12. Juli 2024 wurde noch eine neue Variante, den

Roboterarm mit 3 Gelenken zu steuern, hinzugefügt, sodass die Maus mit dem Endeffektor überlappt und man die volle Kontrolle bei der Steuerung hat. Daneben lag der Fokus auf der Dokumentation und dem Abschlussbericht, in dem die Implementierung, die Ergebnisse und das Feedback aus der Präsentation zusammengefasst wurden. Die abschließende Bewertung reflektierte die erfolgreiche Umsetzung des Projekts und bot Raum für mögliche Verbesserungsvorschläge für zukünftige Entwicklungen.

## 5 Erweiterungsmöglichkeiten

Das Projekt bietet mehrere Ansatzpunkte für zukünftige Erweiterungen. Eine wesentliche Möglichkeit besteht in der Erweiterung der Konfigurationsmöglichkeiten des Roboterarms. Derzeit können Benutzer grundlegende Parameter anpassen, um die Kinematik des Arms zu erforschen. Zukünftige Entwicklungen könnten es ermöglichen, noch detailliertere Aspekte der Armkonfiguration zu verändern. Dazu könnten erweiterte Optionen zur Feinabstimmung der Gelenkwinkel und Geschwindigkeiten gehören sowie zusätzliche Konfigurationsparameter, die eine noch präzisere Anpassung und Untersuchung der Bewegungsabläufe ermöglichen.

Neben den funktionalen Erweiterungen besteht auch die Möglichkeit zur Verbesserung des Designs der Benutzeroberfläche. Die aktuelle Benutzeroberfläche bietet eine funktionale Grundstruktur, jedoch besteht Potenzial für eine ansprechendere und intuitivere Gestaltung. Eine Überarbeitung könnte visuelle Hilfsmittel, wie interaktive Anleitungen und ansprechende Grafiken, beinhalten. Solche Verbesserungen könnten die Benutzerfreundlichkeit erhöhen und die Lernkurve für neue Benutzer verringern, indem sie eine klarere und intuitivere Navigation durch die Simulation ermöglichen.

## 6 Schlussfolgerung

Das entwickelte 2D-Puzzle-Spiel mit Roboterarmsteuerung bietet eine solide Grundlage für die interaktive Simulation von Roboterkinematik. Durch die Möglichkeit, verschiedene Konfigurationen des Roboterarms zu testen, können Benutzer ein tieferes Verständnis für die Funktionsweise und Bewegungsabläufe von Roboterarmen entwickeln. Die flexible Konfigurationsmöglichkeit und

die Drag-and-Drop-Funktionalität haben sich als zentrale Merkmale der Anwendung herausgestellt.

Während der Umsetzung des Projekts haben wir wertvolle Erkenntnisse gewonnen. Besonders hervorzuheben ist, dass die Implementierung eines Roboterarms mit drei Gelenken deutlich komplexer und fehleranfälliger war als ursprünglich erwartet. Die Herausforderungen bei der mathematischen Modellierung und der kinematischen Steuerung haben gezeigt, wie anspruchsvoll die Entwicklung eines solchen Systems sein kann. Bei der Simulation traten immer wieder Schwierigkeiten auf, insbesondere bei der exakten Positionierung der Endeffektoren. Die präzise Berechnung der Gelenkwinkel und deren Koordination erwiesen sich als komplexe Aufgaben, die häufig Anpassungen und Fehlersuche erforderten. Erst durch eine neue Herangehensweise gelang die genaue Positionierung der Endeffektors. Trotz der sorgfältigen Planung und Implementierung sind weiterhin Verbesserungen und Erweiterungen möglich, um die Funktionalität und Benutzerfreundlichkeit weiter zu optimieren. Die gesammelten Erfahrungen haben unsere Kenntnisse in der Roboterkinematik und der praktischen Anwendung von Simulationstechniken erheblich erweitert.

## **7 Implementierung**

### **7.1 Gegenstand der Entwicklung**

In einer zweidimensionalen Umgebung soll ein Puzzlespiel entstehen, das mit Hilfe eines Roboterarms gesteuert wird. Dieser Arm kann per Drag-and-Drop gesteuert werden und so die einzelnen Puzzleteile erreichen und aufheben. Die Anzahl der Gelenke und die Länge der einzelnen Armteile sind dabei über ein Menü variabel einstellbar. Sobald das Puzzle gelöst ist, kann es mit Hilfe einer Korrekturfunktion überprüft werden.

### **7.2 Roboterarm**

Um den Roboterarm später steuern zu können, muss dieser zunächst in einer eigenen Klasse konfiguriert werden. Dabei wird er mit Startwerten versehen und erhält Methoden, die die Positionen des Roboterarms bei der Bewegung

neu berechnen.

### 7.2.1 Initialisierung

Um später beim Starten des Programms einen Roboterarm sehen zu können, muss dieser initialisiert werden. Dabei werden die Anzahl der Gelenke auf zwei sowie die Winkel der drei möglichen Gelenke auf 0 Grad festgelegt. Die Länge der einzelnen Armteile wurde zuvor als Konstante festgelegt auf jeweils 150 eingestellt.

### 7.2.2 Berechnung Winkel

Während der Arm bewegt wird, werden dauerhaft die Winkel in den Gelenken neu berechnet, um den Endeffektor erreichen zu können. Benötigt werden dafür die Differenzen der x- und y-Koordinaten zwischen dem Endeffektor und dem Schultergelenk beziehungsweise dem Ursprung. Mit diesen kann durch den Satz des Pythagoras die Distanz bestimmt werden, die zur Berechnung der Winkel notwendig ist.

$$c = \sqrt{a^2 + b^2}$$
$$c = \sqrt{80^2 + 60^2} = 100$$

Für die Berechnung bei 1, 2 und 3 Gelenken ist unterschiedlicher Code erforderlich. Dieser wird je nach Bedarf durch eine if-else-Anweisung aufgerufen.

Berechnung für 1 Gelenk Handelt es sich um nur ein Gelenk, soll der Endeffektor des Arms in Richtung des Zielpunkts bewegt werden. Für die genauen Koordinaten wird die zuvor berechnete Distanz und der Winkel des Armsegments benötigt. Diesen erhält man durch Einsetzen der Koordinaten in die Arcustangens-Funktion.

$$\theta = \arctan\left(\frac{B}{A}\right)$$
$$\theta = \arctan\left(\frac{60}{80}\right) = 0,64(36,87^\circ)$$

Multipliziert man nun die Armlänge mit dem Sinus beziehungsweise dem Kosinus des Winkels, erhält man den Endeffektor.

$$x = c * \cos(\theta)$$

$$y = c * \sin(\theta)$$

$$x = 100 * \cos(0,64) = 80$$

$$y = 100 * \sin(0,64) = 60$$

Berechnung für 2 Gelenke Bei der Berechnung der Winkel mit zwei Armsegmenten wird wie bei einem Armsegment auch zuerst die Distanz zwischen dem Endeffektor und dem Ursprung berechnet. Zusätzlich sind die Längen der beiden Armsegmente gegeben, woraus ein Dreieck mit drei bekannten Seitenlängen entsteht.

$$a = 50$$

$$b = 40$$

$$c = 63,25$$

Für die Berechnung des Ellenbogenwinkels kann nun der Kosinussatz umgestellt nach dem Winkel angewendet werden.

$$\cos(\theta) = \frac{a^2 + b^2 - c^2}{2 * a * b}$$

$$\cos(\theta) = \frac{50^2 + 40^2 - 63,25^2}{2 * 50 * 40}$$

$$\theta = \arccos(0,025) = 89,86^\circ$$

Nachdem der Ellenbogenwinkel berechnet wurde, kann mit dessen Hilfe auch der Schulterwinkel berechnet werden, indem folgende Formel verwendet wird.

$$\alpha = \arctan\left(\frac{y}{x}\right) - \arctan\left(\frac{b * \cos(\theta)}{a + b * \sin(\theta)}\right)$$

$$\alpha = \arctan\left(\frac{20}{60}\right) - \arctan\left(\frac{40 * \cos(89,86^\circ)}{50 + 40 * \sin(89,86^\circ)}\right)$$

$$\alpha = 18,43^\circ - 37,97^\circ = -19,55^\circ$$

Nachdem beide Winkel berechnet wurden, kann mit Hilfe der Armlängen der Arm konstruiert werden. Die Positionen des Ellenbogengelenks und des Endeffektors werden durch Einsetzen der Winkel in Sinus- und Kosinusfunktionen multipliziert mit der Armlänge bestimmt.

Ellenbogengelenk:

$$x1 = a * \cos(\theta)$$

$$y1 = a * \sin(\theta)$$

$$x1 = 50 * \cos(-19,55^\circ) = 47,12$$

$$y1 = 50 * \sin(-19,55^\circ) = 16,73$$

Endeffektor:

$$x2 = x1 + b * \cos(\theta + \alpha)$$

$$y2 = y1 + b * \sin(\theta + \alpha)$$

$$x2 = 47,12 + 40 * \cos(-19,55^\circ + 89,86^\circ) = 60,6$$

$$y2 = 16,73 + 40 * \sin(-19,55^\circ + 89,86^\circ) = 54,4$$

Diese Punkte werden miteinander verbunden und werden dauerhaft aktualisiert, wodurch der Roboterarm dargestellt wird und sich den Bewegungen anpasst.