



# Bericht Abschlussprojekt

Softwaredesign

Wintersemester 2024/2025

## **Kinematik-Simulation von Ebenen Mechanismen**

Bachelor - Mechatronik, Design und Innovation

Jahrgang: BA-MECH-23

Lehrveranstaltungsleiter: Prof. Dr. Julian Huber, Matthias Panny MSc

Studierende: Andre Muther, Günter Steininger

25. Februar 2025

# Einleitung

In diesem Bericht wird die Umsetzung einer Kinematik-Simulation für ebene Mechanismen beschrieben, die im Rahmen des Softwaredesign-Projekts entwickelt wurde. Ziel des Projekts ist die Entwicklung einer Web-Anwendung zur Simulation der Kinematik von Mechanismen unter Verwendung der Programmiersprache Python und des Frameworks Streamlit. Der Bericht behandelt sowohl die Erfüllung der minimalen Anforderungen als auch die Implementierung zusätzlicher Erweiterungen. Zudem wird eine Analyse der aufgetretenen Fehler sowie deren Lösungen vorgestellt.

# Inhaltsverzeichnis

<b>1</b>	<b>Minimalanforderungen</b>	<b>1</b>
<b>2</b>	<b>Erweiterungen</b>	<b>1</b>
<b>3</b>	<b>Fehleranalyse</b>	<b>2</b>

# 1 Minimalanforderungen

Im Rahmen des Projekts wurde eine Anwendung zur Simulation von ebenen Mechanismen entwickelt, die den folgenden minimalen Anforderungen entspricht:

- **Web-UI mit Streamlit:** Eine Benutzeroberfläche zur Interaktion mit der Mechanismus-Simulation wurde bereitgestellt. Es können Mechanismen definiert und deren Kinematik simuliert werden.
- **Mechanismus-Definition:** Beliebige ebene Mechanismen mit Drehgelenken und starren Gliedern können definiert werden. Jedes Mechanismus besteht aus Gelenken (Punkten) und Gliedern (Stangen), wobei ein Gelenk fixiert wird und als Antrieb fungiert.
- **Kinematikberechnung und Visualisierung:** Für einen definierten Winkelbereich von 0 bis 360 Grad wird die Kinematik des Mechanismus berechnet. Diese Kinematik wird in der Web-Oberfläche visualisiert, wobei die Bahnkurven der Punkte als CSV-Datei exportiert werden können.
- **Speichern und Laden von Mechanismen:** Der Mechanismus sowie seine Kinematik können gespeichert und zu einem späteren Zeitpunkt wieder geladen werden. Die Daten werden im JSON-Format abgelegt.
- **Optimierung:** Das Optimierungsproblem, bei dem die Längenfehler der Glieder minimiert werden, ist implementiert. Dies erfolgt durch den Einsatz der Fehleranalyse und Optimierungsmethode (`scipy.optimize.least_squares`).
- **Validierung des Mechanismus:** Es wird überprüft, ob der definierte Mechanismus valide ist, indem sichergestellt wird, dass mindestens ein Gelenk fixiert wird und alle Gelenke korrekt miteinander verbunden sind.

Die wichtigsten Module und Funktionen, die diese Anforderungen erfüllen, befinden sich in der Datei `visualisierung.py`, welche die Mechanismus- und Kinematik-Logik implementiert, sowie in der Datei `fehleranalyse.py`, die die Berechnung der Längenfehler übernimmt. Der Mechanismus wird in der Datei `mechanismus.py` erstellt und validiert.

# 2 Erweiterungen

Neben den minimalen Anforderungen wurden folgende Erweiterungen umgesetzt:

- **Strandbeest-Mechanismus:** Ein Mechanismus, der einem SStrandbeestBein nachempfunden ist, wurde entwickelt. Diese Erweiterung umfasst die Definition von Gelenken und Verbindungen, die zu einem komplexeren Mechanismus führen, welcher ebenfalls in der Anwendung simuliert werden kann.
- **Schubkurbel-Simulation:** Eine Schubkurbel-Simulation wurde implementiert, bei der ein Mechanismus mit Schubgelenken simuliert wird. Diese Erweiterung fügt der Anwendung einen zusätzlichen Freiheitsgrad hinzu.
- **Fehleranalyse und Visualisierung:** Eine Fehleranalyse wurde implementiert, die die Längenfehler der Glieder in Bezug auf den Drehwinkel berechnet und visuell darstellt. Diese Erweiterung ermöglicht eine genauere Überwachung der Genauigkeit des Mechanismus im Hinblick auf den definierten Drehwinkel.

- **Export und Import von Daten:** Der Export der Kinematik-Daten in das CSV-Format und der Import von gespeicherten Mechanismus-Definitionen bieten den Nutzern eine einfache Möglichkeit, ihre Modelle zu speichern und wiederzuverwenden.
- **Optimierung der Gliederlängen:** Eine Erweiterung ermöglicht es, die Gliederlängen so zu optimieren, dass eine gewünschte Bahnkurve erreicht wird. Dies wurde als zusätzliche Funktion in die Simulation integriert, um die Effizienz und Genauigkeit des Mechanismus zu maximieren.

Diese Erweiterungen stellen sicher, dass die Anwendung nicht nur die grundlegenden Anforderungen erfüllt, sondern auch erweiterte Simulations- und Analysemöglichkeiten für komplexere Mechanismen bietet.

### 3 Fehleranalyse

Trotz intensiver Bemühungen konnten einige Probleme nicht vollständig gelöst werden.

- **Streamlit-Fehler:** Die Anwendung läuft lokal, jedoch konnte der Fehler beim Deployment auf Streamlit trotz mehrfacher Überprüfung der Konfiguration und Abhängigkeiten nicht behoben werden.
- **Unstrukturierte Ordnerstruktur:** Die Ordnerstruktur weist teilweise Redundanzen auf, die die Übersichtlichkeit erschweren. Wir haben uns dazu entschlossen dies so beizubehalten, da diese Struktur eine höhere Flexibilität bei der zukünftigen Erweiterung und Integration neuer Funktionalitäten ermöglicht. Dies haben wir im Hinblick auf die Erfüllung von zusätzlichen Anforderungen so gemacht.
- **Speichern der Mechanismus-Konfiguration:** Der Mechanismus des Strandbeest-Beins wurde als Funktion implementiert. Eine Speicherung als Konfiguration wurde in mehreren Ansätzen geprüft, jedoch nicht effizient umgesetzt.
- **Doppelte Dateifunktionen:** Einige Funktionen waren redundant in verschiedenen Dateien vorhanden. Eine Vereinheitlichung wurde angestrebt, konnte jedoch aufgrund der Projektstruktur nicht vollständig abgeschlossen werden.