



HOCHSCHULE HEILBRONN

Fakultät für Mechanik und Elektronik

---

## Ausgewählte Kapitel der Robotik

# Abschlussprojekt zur Vorlesung

Gruppe 4

**Dozent:** Prof. Andreas Hoch  
**Betreuer:** B.Eng. Fabian Finkbeiner

**Autoren:** Semyon Kondratev 207612  
Lisa-Franziska Schäfer 199318

---

Abgabedatum: 21. Februar 2022

# Inhaltsverzeichnis

<b>1</b>	<b>Einleitung</b>	<b>1</b>
<b>2</b>	<b>Modellbildung des UR10</b>	<b>2</b>
2.1	Download und Konvertierung der step-Dateien . . . . .	2
2.2	Modellierung in Simscape . . . . .	2
2.3	Integration eines Greifersystems und einer Roboter-Halterung . . . . .	3
2.4	Festlegung der Roboterkoordinatensysteme nach Denavit-Hartenberg-Konvention	4
<b>3</b>	<b>Darstellung einer Bewegungssequenz</b>	<b>4</b>
<b>4</b>	<b>Ausgabe von Drehmomentverläufen</b>	<b>6</b>
<b>5</b>	<b>Vergleichsrechnung mit Newton-Euler-Verfahren</b>	<b>6</b>
	<b>Literaturverzeichnis</b>	<b>7</b>

# 1 Einleitung

Im Rahmen des Abschlussprojekts zur Vorlesung Ausgewählte Kapitel der Robotik sollen verschiedene Aufgaben rund um den Roboter UR10 der Firma Universal Robots bearbeitet werden. Zunächst soll das Modell des Roboters in Matlab Simscape nachgebildet werden. Dazu können die step-Dateien des Roboters verwendet werden, welche auf einigen Webseiten als kostenloser Download zur Verfügung stehen. Die Massen und Trägheiten der Armteile bzw. Gelenke sollen dabei abgeschätzt werden. Als Nächstes soll die zugewiesene Bewegungssequenz aus dem Beispielvideo dargestellt werden. Die Bewegung umfasst dabei eine etwa drei Sekunden lange „*Point to Point*“ Bewegung des UR10. Eine weitere Aufgabe ist, die Drehmomentverläufe der Antriebe des Modells in Simscape auszugeben. Als Letztes sollen die von der Simulation berechneten Drehmomente mittels einer Vergleichsrechnung mit Newton-Euler-Verfahren überprüft werden.

Beim UR10 handelt es sich um einen 6-Achs Roboterarm, welcher in der Abbildung 1 zu sehen ist. Sein Aufbau ist an einen menschlichen Arm angelehnt, wodurch er nahezu jede Position in seinem Arbeitsraum erreichen kann. Er ist der größte der UR-Familie und kann Lasten von bis zu 10 kg bei einem Arbeitsradius von bis zu 1300 mm bewegen. Dies ermöglicht ihm ein breites Anwendungsspektrum in der Maschinenbestückung, Palettierung und Verpackung.[Uni]



**Abbildung 1:** UR10 der Firma Universal Robots, Quelle: [https://www.universal-robots.com/3d/images/slider/ur10/small\\_images/rendersd\\_00009.jpg](https://www.universal-robots.com/3d/images/slider/ur10/small_images/rendersd_00009.jpg)

## 2 Modellbildung des UR10

Für die erste Aufgabe soll der Roboter in der Matlab Simulationssoftware Simscape modelliert werden. Anders als in der Aufgabenstellung angegeben, existiert jedoch nur eine einzelne step-Datei über den gesamten Roboter als Download, welche alle Armteile enthält.

### 2.1 Download und Konvertierung der step-Dateien

Die step-Datei des UR10 wird auf diversen Webseiten kostenlos zum Download angeboten. Für das Projekt wurde die Datei der Firma „SG-Automatisierungstechnik GmbH“<sup>1</sup> verwendet. Die step-Datei lässt sich jedoch nicht direkt in Matlab einlesen, sondern muss mittels des „*Simscape Multibody Link Plugin*“ konvertiert werden. Dabei handelt es sich um ein Zusatzprogramm, welches in bestimmten CAD Anwendungen installiert werden kann und den Export von XML- und Geometriedateien ermöglicht [Mat]. Als CAD Anwendung für die Konvertierung wurde SolidWorks gewählt. Der Datelexport aus SolidWorks erzeugte eine XML-Datei, eine Geometriedatei „*UR10\_DataFile.m*“ sowie acht step-Dateien der verschiedenen Armteile und der Basis.

### 2.2 Modellierung in Simscape

Nun kann die XML-Datei mittels dem Befehl „*sm\_import()*“ in Matlab Simscape eingelesen und mit den step-Dateien der Roboterbauteile verknüpft werden. Nach einer kurzen Sichtung des Modells stellte sich heraus, dass die Transformationen zwischen den Bauteilen zwar korrekt, jedoch die interne Verknüpfung des Blockdiagramms nicht stimmig war. Beim Einfügen von Aktuatoren bewegte sich ausschließlich das angesteuerte Armteil, während der Rest der kinematischen Kette statisch in der Ausgangsposition verblieb.

Aus diesem Grund wurde der Import einer URDF-Datei des Roboters versucht, was ebenso über den Befehl „*sm\_import()*“ möglich ist. Die verwendete Datei kann aus dem Github-Repository von „*Positronics Lab*“<sup>2</sup>, einer Forschungsorganisation der George Washington Universität, kostenlos heruntergeladen werden. Hierbei erzeugt Matlab eine eigene XML-Datei, in welcher das Projekt gespeichert ist. Mit Einfügen der step-Dateien der Armteile erhält man ein funktionierendes Robotermodell des UR10. Der originale Output nach Einlesen der URDF-Datei ist in der Abbildung 2 zu sehen. Die Massen und Trägheiten der Armteile wurden aus der zuvor erhaltenen Geometriedatei entnommen.

---

<sup>1</sup><https://www.sg-automation.at/1904594/Downloads-UR10>

<sup>2</sup>[https://github.com/PositronicsLab/reveal\\_packages/blob/master/industrial\\_arm/scenario/models/urdf/ur10/ur10.urdf](https://github.com/PositronicsLab/reveal_packages/blob/master/industrial_arm/scenario/models/urdf/ur10/ur10.urdf)

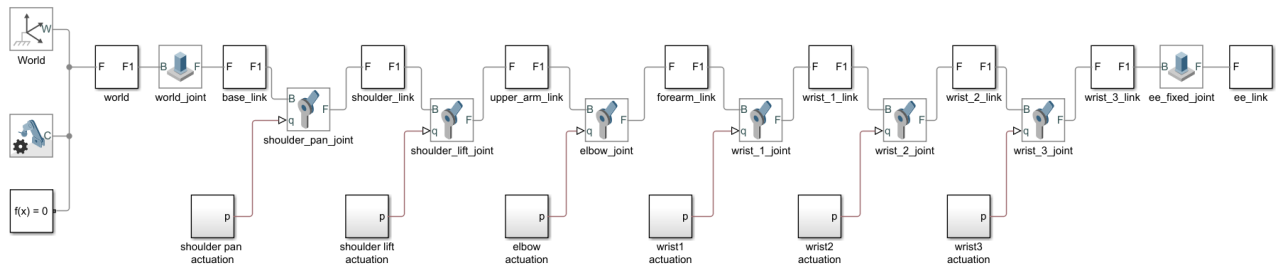


Abbildung 2: Originales Blockdiagramm des UR10 aus der URDF-Datei

## 2.3 Integration eines Greifersystems und einer Roboter-Halterung

Um dem gezeigten Roboter aus dem Beispielvideo möglichst nahe zu kommen, wurde zusätzlich eine Halterung und ein Greifersystem konstruiert. Die Halterung besteht aus einem quaderförmigen Block, bei dem eine Kante nach oben verschoben ist. Dadurch entsteht eine schiefe Ebene, auf der das Robotermodell wie im Video montiert werden kann.

Das Greifersystem ist aus mehreren Teilen zusammengebaut, einer Werkzeughalterung, zwei Greifern und einem Zylinderstift. Die Werkzeughalterung ist ein Prisma mit trapezförmiger Grundfläche. Der Zylinderstift, welcher im Beispielvideo zum Betätigen eines Tasters verwendet wird, ist an einer geraden Fläche an der Haltung befestigt. Für die beiden seitlich angebrachten Greifer wurde die step-Datei eines möglichst ähnlich aussehenden Greifers von der Website „TraceParts“<sup>3</sup> verwendet. Das fertige Robotermodell mit Halterung und Greifersystem ist in Abbildung 3 dargestellt.

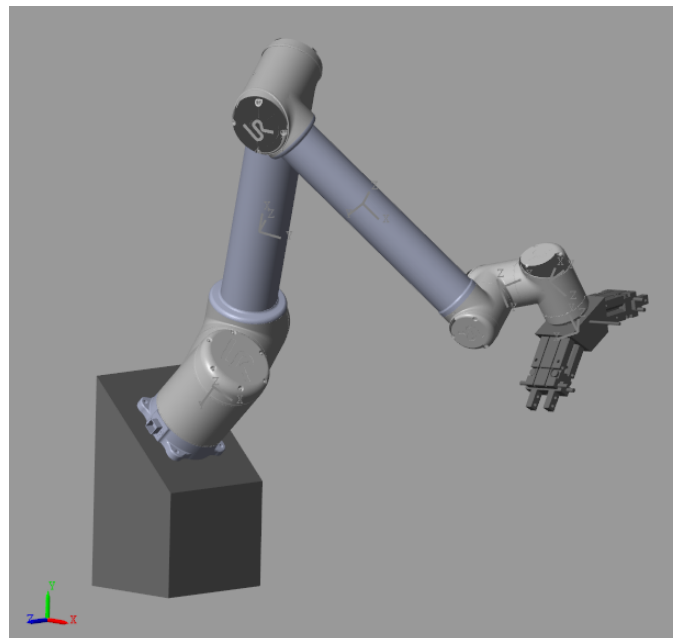
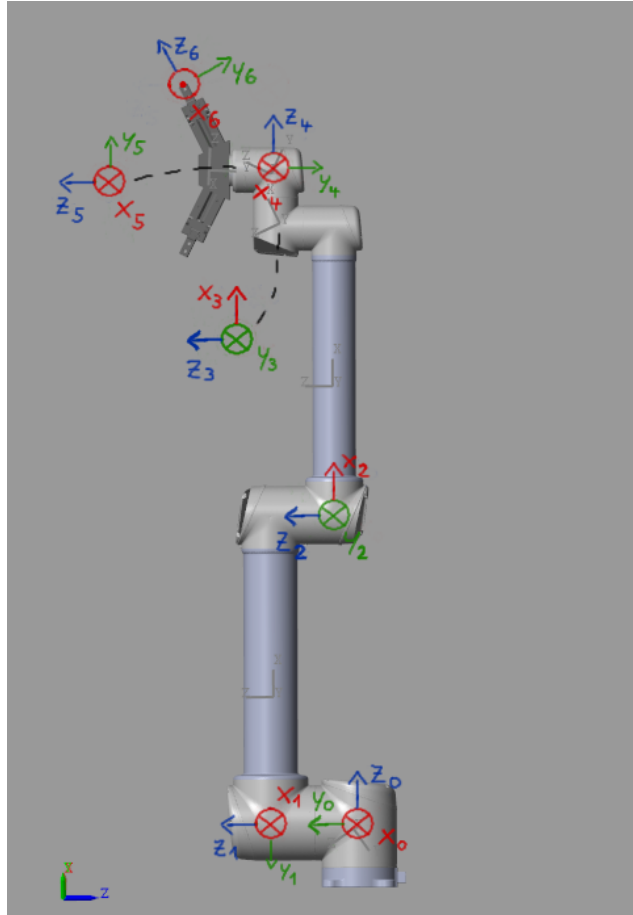


Abbildung 3: UR10 mit Halterung und Greifersystem

<sup>3</sup><https://www.traceparts.com/de/product/apore-2backenparallelgreifer?CatalogPath=APORE%3AAPORE.040.010.010&Product=10-28092012-071699>

## 2.4 Festlegung der Roboterkoordinatensysteme nach Denavit-Hartenberg-Konvention

Für die spätere Vergleichsrechnung mit dem Newton-Euler-Verfahren ist es von Vorteil die Koordinatensysteme der Gelenkachsen des Roboters einheitlich zu definieren. Hierzu bietet sich die Festlegung nach Denavit-Hartenberg-Konvention an. Die nachfolgende Abbildung 4 zeigt die Koordinatensysteme, wie sie auch im Blockdiagramm definiert sind.



**Abbildung 4:** Koordinatensysteme der Gelenke und des TCP nach Denavit-Hartenberg-Konvention

## 3 Darstellung einer Bewegungssequenz

Die geforderte PTP Bewegung des Roboters soll die Zeitstempel 1:05 bis 1:08 im Beispielvideo umfassen. Die daraus resultierende Start- und Endposition ist in den Abbildungen 5a und 5b zusehen. Die Startposition ist mittig an der Türe, die Endposition an der Steuerung auf der rechten Seite. Nach Erreichen der Endposition betätigt der Roboter einen Taster mit seinem am Greifer montierten Zylinderstift und startet dadurch die Werkstückbearbeitung in der Maschine. Diese Tasterbetätigung ist in der Bewegungssequenz ebenfalls umgesetzt.

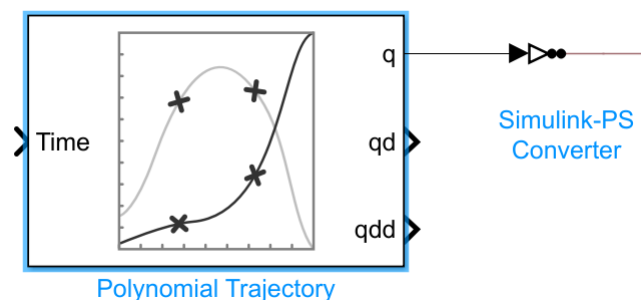


(a) Startposition

(b) Endposition

**Abbildung 5:** Start- und Endposition der PTP Bewegung,  
Quelle: <https://www.youtube.com/watch?v=ugxDtmykE4>

Die Bewegung ist mittels separater Ansteuerung jeder Gelenkachse realisiert. Dazu wird ein Block namens „*Polynomial Trajectory*“ verwendet, welcher Trajektorien durch gegebene Weg- und Zeitpunkte generiert. Der verwendete Block ist in Abbildung 6 auf der linken Seite zu sehen. In unserem Fall handelt es sich bei den Wegpunkten um Winkelstellungen des Gelenks. Daher ist nach dem Trajektoriengenerator ein „*Simulink-PS-Converter*“ eingefügt, der das resultierende einheitenlose Signal des Generators in ein physikalisches mit der Einheit Rad umwandelt.



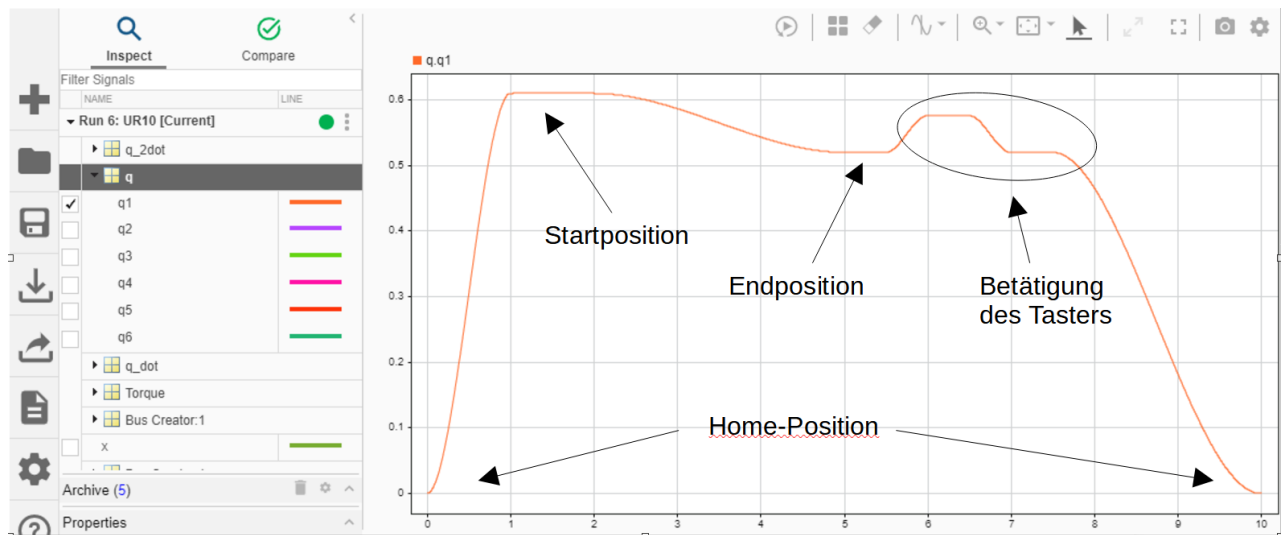
**Abbildung 6:** Blockdiagramm des Aktuators des ersten Gelenks

Für die erste Gelenkachse sind die folgenden Zahlenwerte für die Weg- und Zeitpunkte eingetragen:

*Wegpunkte* :  $[0, 0.61, 0.61, 0.52, 0.52, 0.576, 0.576, 0.52, 0.52, 0]$

*Zeitpunkte* :  $[0, 1, 2, 5, 5.5, 6, 6.5, 7, 7.5, 10]$

Im „*data\_inspector*“ von Matlab Simulink kann das erzeugte Signal des Trajektoriengenerators eingesehen werden. Die Abbildung 7 zeigt beispielhaft wieder die erste Gelenkachse. Zum erleichterten Verständnis sind die verschiedenen Positionen und Aktionen des Roboters im Signal markiert.



## 4 Ausgabe von Drehmomentverläufen

## 5 Vergleichsrechnung mit Newton-Euler-Verfahren



## Literaturverzeichnis

- [Mat] MATHWORKS: *Install the Simscape Multibody Link Plugin - MATLAB & Simulink - MathWorks Deutschland.* <https://de.mathworks.com/help/physmod/smlink/ug/installing-and-linking-simmechanics-link-software.html>, Abruf: 21.02.2022
- [Uni] UNIVERSAL ROBOTS (GERMANY) GMBH: *Mit Cobots Ihre Produktion automatisieren / Universal Robots.* <https://www.universal-robots.com/de/produkte/>, Abruf: 21.02.2022