UR Roboter

Das Ziel dieses Praktikums ist es, die wichtigsten Befehle des UR Roboters kennenzulernen und damit die theoretischen Grundlagen der Kinematik zu verdeutlichen.

Am Ende soll der Roboter in der Lage sein, eine geometrische Figur (Viereck) zu zeichnen.

Schreiben Sie die folgenden Programme:

Programm 1: Hello World
siehe „The First Program“ in der UR Dokumentation

Programm 2: Drehung um Stiftspitze
Kalibrieren Sie möglichst genau den Stift als TCP. Überprüfen Sie Ihre Kalibrierung indem Sie Drehungen um die Stiftspitze durchführen.

Programm 3: Viereck im Raum mit Polyscope
Tipp: es sollte ein Annäherungspunkt angefahren werden bevor der erste Referenzpunkt des Vierecks angefahren wird

Validierung der programmierten Vorwärtskinematik

Überprüfen Sie die Werte (x,y,z,rx,ry,rz) der Vorwärtskinematik aus der selbstprogrammierten Python Bibliothek mit den Werten aus der Robotersteuerung.
Führen Sie diese Überprüfung einmal mit und einmal ohne eingestellten Stift als TCP durch.

myCobot 280 Roboter

https://docs.elephantrobotics.com/docs/gitbook-en/2-serialproduct/2.1-280/2.1.6.1-IntroductionOfProductParameters.html

Stellen Sie die Tabelle mit den Denavit-Hartenberg-Parametern auf und implementieren Sie die Vorwärtskinematik. Überprüfen Sie die Ergebnisse auf Plausibilität.

RoArm-M1 Roboter

https://www.waveshare.com/wiki/RoArm-M1

Stellen Sie die Tabelle mit den Denavit-Hartenberg-Parametern auf und implementieren Sie die Vorwärtskinematik. Überprüfen Sie die Ergebnisse auf Plausibilität.

Tipps zur Ansteuerung des myCobot 280

Installation der Python Bibliothek pymycobot

```
# Example
from pymycobot import MyCobot280

mc = MyCobot280('COM3', 115200)

#print(mc.get_system_version())

# Gets the current angle of all joints
angles = mc.get_angles()
print(angles)

# Set 1 joint to move to 40 and speed to 20
#mc.send_angle(4, 30, 20)

#mc.sync_send_angles([0,0,0,0,0,0],20)

#q = [60,-20,-30,-40,50,60]
q = [0,0,0,0,0,0]
mc.sync_send_angles(q, 20)
```

Tipps zur Ansteuerung des RoArm-M1

Die Kommunikation erfolgt über JSON Objekte, die über die serielle Schnittstelle gesendet werden. Es kann z.B. der serielle Monitor der Arduino-IDE verwendet werden.

Abfrage der Gelenkwinkel:

```
{"T":5}
```

Fahrbefehl:

```
{"T":1,"P1":180,"P2":0,"P3":90,"P4":90,"P5":180,"S1":200,"S2":200,"S
3":200,"S4":200,"S5":200,"A1":60,"A2":60,"A3":60,"A4":60,"A5":60}
```