1	2	3	4	5	$\sum$

Prof. Daniel Göhring

Robotik, WiSe 18/19

Übung 03

Dominik Dreiner, Mai-Phú Pham, Yichi Chen

---

1 Simples Einparken (7 Punkte)

- Der Link für das von uns aufgenommene Video:
  `https://git.imp.fu-berlin.de/phup/robotik-uebungen/blob/master/assignment_3/einpark_video.mp4`
- Der Link für unseren *parking_maneuver.py*:
  `https://git.imp.fu-berlin.de/phup/robotik-uebungen/blob/master/assignment_3/src/simple_parking_maneuver/src/parking_maneuver.py`
- Der Link für unseren *drive_control.py*:
  `https://git.imp.fu-berlin.de/phup/robotik-uebungen/blob/master/assignment_3/src/simple_drive_control/src/drive_control.py`

2 (1 Punkt)

$$ {}^B_A T = \begin{pmatrix} \cos\alpha & -\sin\alpha & 0 & -1 \\ \sin\alpha & \cos\alpha & 0 & 4 \\ 0 & 0 & 1 & 5 \\ 0 & 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} 0 & -1 & 0 & -1 \\ 1 & 0 & 0 & 4 \\ 0 & 0 & 1 & 5 \\ 0 & 0 & 0 & 1 \end{pmatrix}. $$

3 (1 Punkt)

- Da $\theta = -\frac{3\pi}{2}$ ist, haben wir zunächst

$$ \begin{array}{lllll} \varepsilon_0 & = & \cos\frac{\theta}{2} & \approx & -0.707, \\ \varepsilon_1 & = & k_X \sin\frac{\theta}{2} & = & 0, \\ \varepsilon_2 & = & k_Y \sin\frac{\theta}{2} & = & 0, \\ \varepsilon_3 & = & k_Z \sin\frac{\theta}{2} & \approx & -0.707. \end{array} $$

  Daraus ergibt sich die Matrix für Rotation:

$$ \begin{aligned} {}^A_B R &= \begin{pmatrix} 1-2\varepsilon_2^2-2\varepsilon_3^2 & 2(\varepsilon_1\varepsilon_2-\varepsilon_3\varepsilon_0) & 2(\varepsilon_1\varepsilon_3+\varepsilon_2\varepsilon_0) \\ 2(\varepsilon_1\varepsilon_2+\varepsilon_3\varepsilon_0) & 1-2\varepsilon_1^2-2\varepsilon_3^2 & 2(\varepsilon_2\varepsilon_3-\varepsilon_1\varepsilon_0) \\ 2(\varepsilon_1\varepsilon_3-\varepsilon_2\varepsilon_0) & 2(\varepsilon_2\varepsilon_3+\varepsilon_1\varepsilon_0) & 1-2\varepsilon_1^2-2\varepsilon_2^2 \end{pmatrix} \\ &= \begin{pmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix}. \end{aligned} $$

  Somit erhalten wir:

$$ x_{neu} =^A_B R \cdot x = (0, 2, 0). $$

- Es gilt zunächst

$$\begin{array}{rcccl} \theta & = & 2 \cdot \arccos \varepsilon_0 & = & 120°, \\ k_X & = & \frac{\varepsilon_1}{\sqrt{1-\varepsilon_0^2}} & \approx & -0.58, \\ k_Y & = & \frac{\varepsilon_2}{\sqrt{1-\varepsilon_0^2}} & \approx & -0.58, \\ k_Z & = & \frac{\varepsilon_3}{\sqrt{1-\varepsilon_0^2}} & \approx & 0.58. \end{array}$$

$(0.5, -0.5, -0.5, 0.5)$ entspricht somit einer Axis mit $(-0.58, -0.58, 0.58)$, und einem Winkel mit $120°$, also $\frac{2\pi}{3}$.

4 (1 Punkt)

$$\begin{aligned} z &= x \times y \\ &= (-\sqrt{0.5}, \sqrt{0.5}, 0) \times (\sqrt{0.5}, \sqrt{0.5}, 0) \\ &= \begin{vmatrix} e_1 & -\sqrt{0.5} & \sqrt{0.5} \\ e_2 & \sqrt{0.5} & \sqrt{0.5} \\ e_3 & 0 & 0 \end{vmatrix} \\ &= (0, 0, -1). \end{aligned}$$

5 **Kreisfahrt** (2 Bonus-Punkte)

- Der Link für das von uns aufgenommene Video (auf Rennbahn):
`https://git.imp.fu-berlin.de/phup/robotik-uebungen/blob/master/assignment_3/zusatzaufgabe/Kreisfahren_Rennbahn_knapp.mp4`
- Der Link für das von uns aufgenommene Video (auf Teppich):
`https://git.imp.fu-berlin.de/phup/robotik-uebungen/blob/master/assignment_3/zusatzaufgabe/Kreisfahren_Teppich_Viertelkreis.mp4`
- Der Link für unseren *parking_maneuver.py*:
`https://git.imp.fu-berlin.de/phup/robotik-uebungen/blob/master/assignment_3/zusatzaufgabe/parking_maneuver.py`
- Der Link für unseren *drive_control.py*:
`https://git.imp.fu-berlin.de/phup/robotik-uebungen/blob/master/assignment_3/zusatzaufgabe/drive_control.py`

Bemerkungen:

1. Unser Testmodelcar bei der Zusatzaufgabe ist die Ada, Nummer 124.
2. Der Radabstand beträgt circa 25 cm.
3. Wir haben den Einschlagwinkel beim Kreisfahren auf 55° gestellt. Da wir doch einen Durchmesserunterschied zwischen Einschlagwinkel 55° und 60° feststellen können, gehen wir davon aus, dass der maximale Einschlagwinkel unseres Modelcars doch größer als 55° ist.
4. Laut einiger Quellen[1] sollte der Einschlagwinkel der folgenden Formel folgen,

$$R = \frac{D}{sin(\alpha)},$$

[1] `https://rechneronline.de/bremsweg/wendekreis.php`

Wobei R dem Radius, D dem Radabstand, α dem Einschlagwinkel jeweils entspricht. Somit ergibt sich nach unserer Rechnung die theoretische $\alpha = 14.5°$, welche sich leider nur stark von der realistischen $\alpha = 55°$ differenziert.

5. Wir haben zuerst mittels des Befehls *pub_forward.publish(1.0)* festgestellt, dass eine Entfernung 1.0 im ROS in der Realität circa 1.2 m entspricht.

6. Nach der folgenden Rechnung ergibt sich der Kreisumfang circa 6.28 m.

$$U = 2\pi R \approx 6.28\ m.$$

Das entspricht wiederum $6.28/1.2 \approx 5.24$ Entfernung beim ROS.

7. Bei unserem Kreisfahren haben wir dementsprechend den Fahrabstand auf 5.2 gesetzt. (s. *parking_maneuver.py* Dann ist das Testergebnis bei einem Rundfahrt tatsächlich knapp eine Runde. (s. *Kreisfahren_Rennbahn_knapp.mp4*[2])

8. Der tatsächliche Radius unserer Kreisfahrt beträgt circa 104 cm.

9. Da es im Labor nicht genügenden Platz auf dem Teppich für eine komplette Kreisfahrt gibt, haben wir die Kreisfahrt stattdessen auf einen Viertelkreis reduziert. (s. *Kreisfahren_Teppich_Viertelkreis.mp4* [3])

10. Unsere Messung für den Radius der Viertelkreisfahrt ist natürlich sehr grob. Sie besagt einen Radius mit circa 103 cm. Also keinen krassen Unterschied zu der Fahrt auf der Rennbahn.

11. Intuitiv würden wir behaupten, dass der Radius auf dem Teppich etwas kleiner als den auf der Rennbahn sein konnte, da es beim Teppich um einen höheren Reibungskoeffizienten handeln sollte.

[2] https://git.imp.fu-berlin.de/phup/robotik-uebungen/blob/master/assignment_3/zusatzaufgabe/Kreisfahren_Rennbahn_knapp.mp4

[3] https://git.imp.fu-berlin.de/phup/robotik-uebungen/blob/master/assignment_3/zusatzaufgabe/Kreisfahren_Teppich_Viertelkreis.mp4