

## 11. Assignment – Obstacle Avoidance

by Pascal Müller, Fabian Casares, Friedrich Müller

[https://github.com/frimuell/catkin\\_ws\\_user/blob/master/tasks\\_wise1819/ub11/](https://github.com/frimuell/catkin_ws_user/blob/master/tasks_wise1819/ub11/)

### Exercise 1 – Calculate Distance to Nearest Obstacle on Lane

#### 1. Transformation der Koordinaten

Die Transformationsmatrix  $T$  sieht wie folgt aus:

$$\begin{pmatrix} \cos(a) & -\sin(a) & 0 & x_{cw} \\ \sin(a) & \cos(a) & 0 & y_{cw} \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

, wobei  $x_{cw}$  und  $y_{cw}$  die Koordinaten des Autos in Respektive zum Welt-Koordinatensystem sind.

Diese wird (in unserem Fall) multipliziert mit einem Vektor  $v$ :

$$\begin{pmatrix} x_{pc} \\ y_{pc} \\ 0 \\ 1 \end{pmatrix} = v$$

, wobei  $x_{pc}$  und  $y_{pc}$  die Koordinaten eines Lidar-Punktes in Bezug zum Auto sind. (Die z-Koordinate kann hier vernachlässigt werden.)

Wir benutzen das Inverse der obigen Transformationsmatrix um Weltkoordinaten in Koordinaten in Respektive zum Auto umzuwandeln.

#### 2.+3. Hindernis-Erkennung in der Spur

Hierfür haben wir unseren „Zielpunkt“ des Autos verwendet und dessen Koordinaten (in Respektive zur Welt) transformiert, sodass dieser in Respektive zum Auto vorliegt. Anschließend haben wir stets alle vom Lidar wahrgenommenen Punkte betrachtet. Lagen davon zu irgendeinem Zeitpunkt während der Fahrt eine Treshold Anzahl von Punkten im Radius von 15cm des angestrebten Punktes in der Trajektorie, wurde dies als Hindernis erkannt und ein Spurenwechsel initiiert. (Dies setzt die Annahme voraus, dass während der Fahrt keine Veränderungen der Hindernisse geschieht, also bewegliche Hindernisse ausgeschlossen sind.)

Code:

[https://github.com/frimuell/catkin\\_ws\\_user/blob/master/tasks\\_wise1819/ub11/drive\\_on\\_line\\_with\\_l oc.py](https://github.com/frimuell/catkin_ws_user/blob/master/tasks_wise1819/ub11/drive_on_line_with_l oc.py)

Video:

[https://github.com/frimuell/catkin\\_ws\\_user/blob/master/tasks\\_wise1819/ub11/Exercise%201.mp4](https://github.com/frimuell/catkin_ws_user/blob/master/tasks_wise1819/ub11/Exercise%201.mp4)

#### 4. Scannen der zweiten Spur im Falle eines Spurenwechsels

Wird ein Spurenwechsel initiiert, so wird gleichzeitig die zukünftige lane geprüft. Unsere Idee war zeitgleich mit dem Erkennen eines Hindernisses die zweite Fahrbahn innerhalb des nächsten Meters zu scannen und anzuhalten, sollte dort ebenfalls ein Hindernis gefunden werden.