

1	2	Σ

Prof. Daniel Göhring
Robotik, WiSe 18/19

Übung 08: Velocity Control and Lateral Control

Dominik Dreiner, Mai-Phú Pham, Yichi Chen

1 Velocity controller (5 Punkte)

Unser Code für die erste Aufgabe befindet sich in folgendem Link.

https://git.imp.fu-berlin.de/phup/robotik-uebungen/blob/master/Aufgaben/aufgabe08/catkin_ws_user/src/image_processing/src/line_driving.py

Unsere Ergebnisse zeigen sich wie folgt.

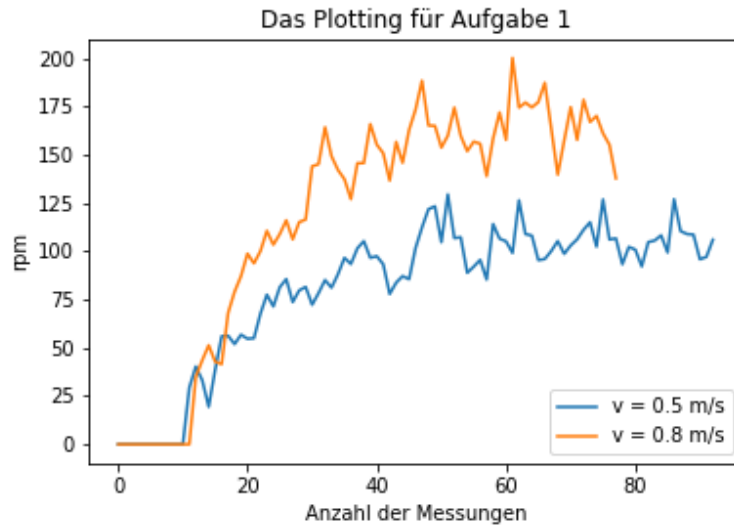


Abbildung 1: Die rpm-Veränderungen nach der Zeit

Dabei handelt es sich um circa 6 Messungen pro Sekunde. Wir hätten natürlich in unserer Funktion auch `time.time()` verwenden können, um die konkrete Zeitmessung zu vermitteln.

Bei unserem PID-Kontroller haben wir erst mal nur die P und D-Kontroller umgesetzt, indem wir uns $K_P = 0.3$ und $K_D = 0.2$ nehmen. Dabei dient der D-Kontroller für die Dämpfung. Dabei haben wir die folgende Formel als Ersatz für die Ableitung von v eingesetzt.

$$\dot{v} = \frac{v_{\text{current}} - v_{\text{last}}}{t_{\text{current}} - t_{\text{last}}} = \frac{\Delta v}{\Delta t}. \quad (1)$$

2 Control a car on an oval circuit (5 Punkte)

Die Aufgabe besteht dabei eigentlich aus zwei Hauptteilen:

1. den *Velocity controller* in unserem älteren Code für den Steuerkontroller zu integrieren.
2. die Kurve zu erkennen, damit das Modelcar auf einer Kurve mit einer Geschwindigkeit v_2 fährt, hingegen auf einer geraden Linie mit v_1 , wobei $v_2 < v_1$ gilt. Mit den zwei unterschiedlichen Geschwindigkeiten können wir eigentlich unsere Ergebnissen von der Abb. 1. direkt anwenden.

Jedoch liegt der Knackpunkt der zweiten Aufgabe darin, die Kurve richtig zu erkennen. Dabei haben wir uns die folgenden 3 unterschiedlichen Methoden ausgedacht.

1. **Primitive:** Da wir wissen, dass das Modelcar dabei gegen den Uhrzeigersinn fahren wird, setzen wir einfach eine Schwelle für den Steuerwinkel, z.B. $s = -20^\circ$. Wenn der aktuelle Steuerwinkel α beim Fahren kleiner als s wäre, wird die Zielgeschwindigkeit auf v_2 gesetzt. Ansonsten auf v_1 .
2. **Punktenmethode:** Finden wir zwei der vertikalen Mittellinie am nächsten befindende weiße Pixel, die sich jeweils auf zwei horizontalen Geraden befinden. Wenn sich der gefundene Pixel der oberen Gerade eh an der linksobigen Ecke des Bildes befindet, ist es dann ein Indiz dafür, dass es sich gerade um eine Kurve handelt. Der gefundene Pixel der unteren Gerade dient wiederum zum Winkelsteuerung, was wir beim letzten Zettel einwandfrei implementiert haben. Allerdings ist es dabei vorausgesetzt, dass die Autobahn ideal erstellt ist. Ansonsten ist die Punktenmethode an sich eh sehr beschränkt.
3. **Linienmethode:** Wenn sich dem Modelcar nahliegenden Tangente bei einer Kurve mittels RANSACs doch feststellen lässt, können wir dann mittels der Steigerung und des Schnidepunkts zur x -Axis die Kurve feststellen.
4. **Kurvenerkennung:** *Mittels polynomialer Interpolation??*

Zugegebenermaßen haben wir es leider nicht rechtzeitig schaffen können, die oben erwähnten Methoden umzusetzen. Mit der primitiven Methode war es wegen technisches Problems der Kamera knapp. Trotzdem fügen wir einen Link hiermit ein. Wenn es welche Videos doch im folgenden Link vorhanden sind, handelt es sich lediglich um eine **nachträgliche** Videohochladung. Daher lassen sie sich ggf. komplett ignorieren.

https:

[//git.imp.fu-berlin.de/phup/robotik-uebungen/tree/master/Aufgaben/aufgabe08/Videos](https://git.imp.fu-berlin.de/phup/robotik-uebungen/tree/master/Aufgaben/aufgabe08/Videos)