Practical Course Project Report

Learning for self driving cars and intelligent systems

Yining Ma, Marc Brede

Technical University Munich

Contents

1 Week One

This week's task is concerned with Model Predictive Control of a vehicle. Using an optimization algorithm, a vehicle is to be guided from a starting position to the desired position. The first step was to come up with the vehicle dynamic equations that model the car's movement. The equations that we have used for this can be seen in the following.

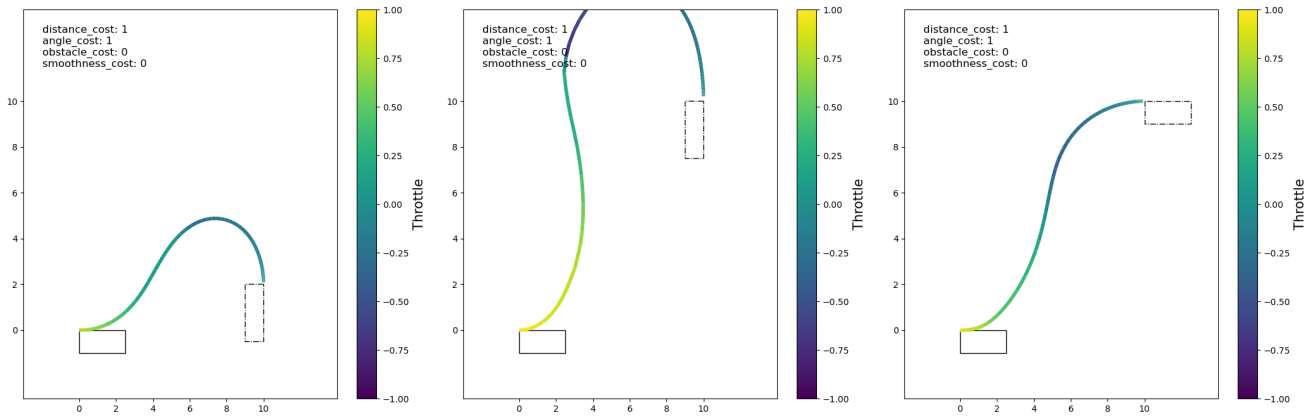
$$x_{t+1} = x_t + v_t \cdot \cos(psi_t) \cdot dt \tag{1}$$

$$y_{t+1} = y_t + v_t \cdot \sin(psi_t) \cdot dt \tag{2}$$

$$psi_{t+1} = psi_t + v_t \cdot \tan(steering) \cdot 0.5 \cdot dt \tag{3}$$

$$v_{t+1} = 0.99 \cdot v_t + pedal \cdot dt \tag{4}$$

We want to optimize for the two variables that we can directly influence: steering and pedal. We came up with a cost function that penalizes the distance and the angle difference from the current position of the vehicle to the desired position, see equation 5. Both of these types of costs have their weighting factor w_{dis} and w_{angle} which we initially set to one. Using different desired end-position p_{des}, figure 1 shows the trajectory and pedal input that the vehicle took during the optimization of the cost function.



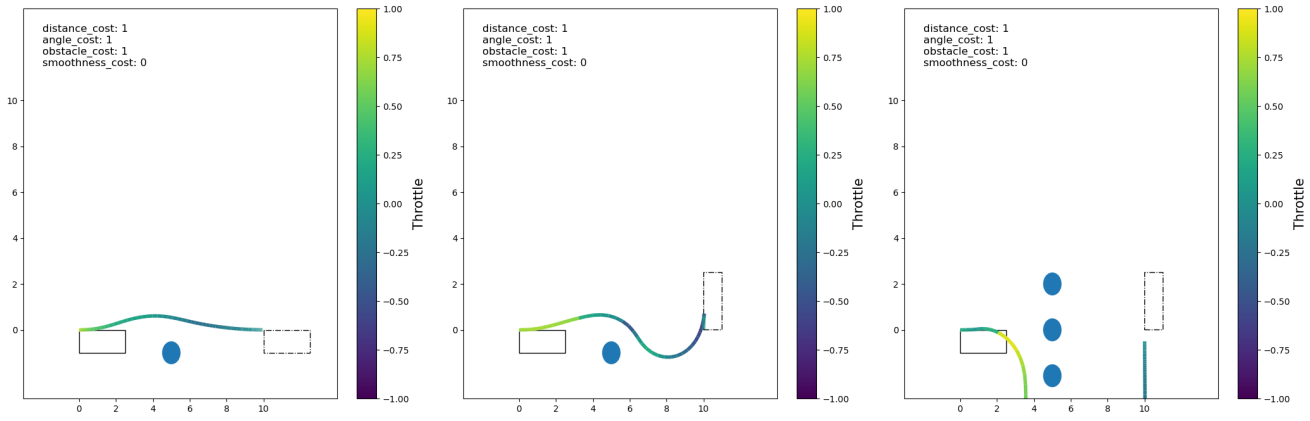
(a) Link to video (b) Link to video (c) Link to video

Figure 1: Trajectory and pedal input visualized for different desired positions

In the next step we introduced obstacles that the vehicle has to go around. If the vehicle gets closer to an obstacle than a minimum distance dis_{min}, it gets penalized by the inverse of its distance to the obstacle times a weighting factor w_{obs}, see 6. The minimum distance dis_{min} was set to 1.5 and the weighting factor w_{obs} to 1. The results can be seen in Figure 2.

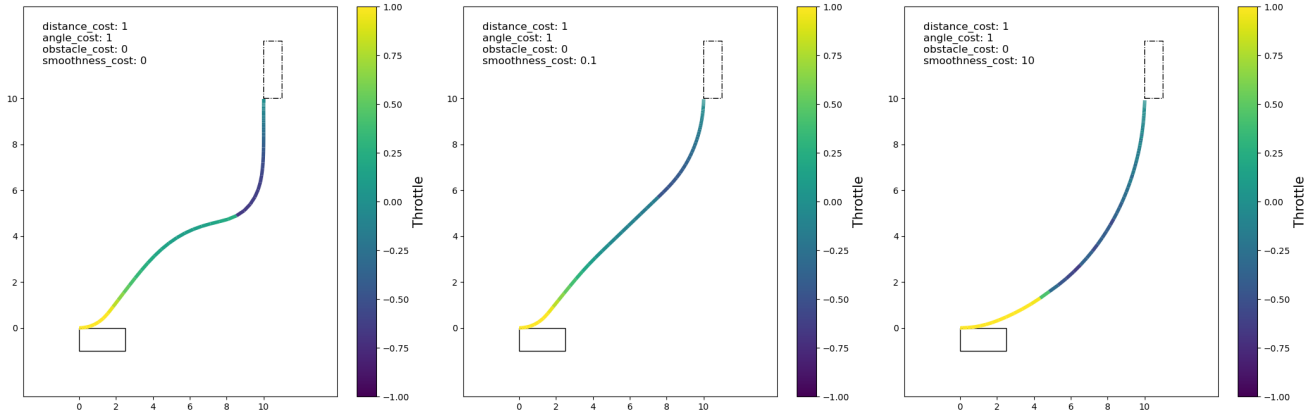
As the final adjustment, we introduced the notion of smoothness, i.e. the perceived comfortableness of the vehicle's passengers. It can either be modeled using the derivative of the vehicle's velocity w.r.t. time, or it can be modeled with the derivative of the pedal input w.r.t. time. The time-derivative of the pedal input is proportional to the change in acceleration, i.e. second time derivative of the vehicle's velocity. We argue that the change in acceleration is the perceived comfortableness of the passengers. The same principle applies to the perceived comfortableness of the steering. Therefore, we modeled the smoothness by penalizing a change

in the steering and the pedal input from t to t+1 with a weighting factor w_{smooth}, see equation 7. Different values for this weight lead to different trajectories, as can be seen in figure 3.



(a) Link to video (b) Link to video (c) Link to video

Figure 2: Vehicle avoiding obstacles while going to the desired position



(a) Link to video (b) Link to video (c) Link to video

Figure 3: Trajectory of the vehicle for different smoothness values

$$cost_1 = \|p_{des} - p_{curr}\|_2 \cdot w_{dis} + \|\angle_{des} - \angle_{curr}\| \cdot w_{angle} \tag{5}$$

$$cost_2 = cost_1 + \sum_o \begin{cases} \|p_o - p_{curr}\|_2^{-1} \cdot w_{obs} & \text{if } \|p_o - p_{curr}\|_2 < dis_{min} \\ 0 & \text{otherwise} \end{cases} \tag{6}$$

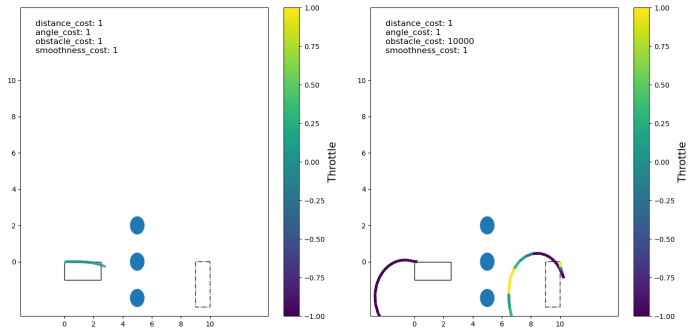
$$cost_3 = cost_2 + \|steering_{t-1} - steering_t\| \cdot w_{smooth} + \|pedal_{t-1} - pedal_t\| \cdot w_{smooth} \tag{7}$$

There are some limitations to these optimizations and whether the algorithm can guide the vehicle to the desired position depends on the setting. Additionally, it highly depends on the weighting factors that we choose and we realized that some weighting factors work better than others. One observation that we made is that the vehicle tends to get stuck in obstacles often. Choosing a weighting factor w_{obs} that is much higher than 1 improved this, see figure 4.

Another parameter that we were able to improve is the smoothness weight w_{smooth}. While higher values yields

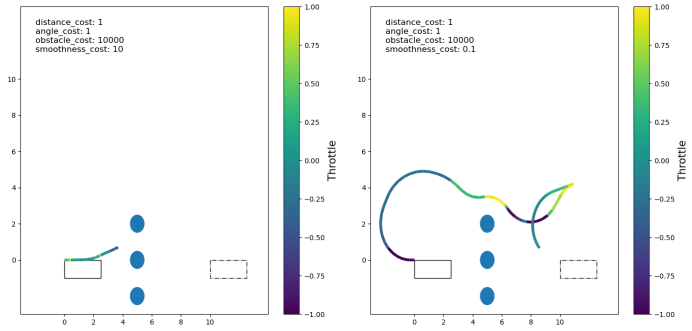
a smoother trajectory, often the algorithm gets stuck in situations in which the vehicle does not move anymore. Choosing a lower value for w_{smooth} improved this problem, see figure 5.

Finally, we have also observed that the optimization can get stuck in local minima if the value for the horizon is not big enough. Often, the optimization algorithm has to take steps against the gradient to get out of local minima. The optimization will only go against the gradient if it finds a position in the search range that yields an even lower value. The higher the horizon, the more likely is the optimization to find such an lower value, see figure 6.



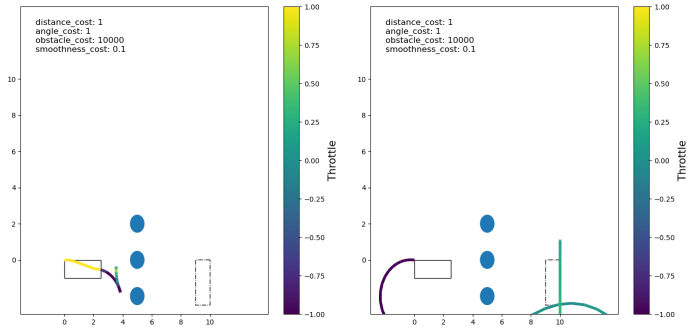
(a) Link to video (b) Link to video

Figure 4: Vehicle getting stuck in obstacle for low values for w_{obs} while for higher values it does not



(a) Link to video (b) Link to video

Figure 5: Vehicle getting stuck/not stuck depending on the value for w_{smooth}



(a) Link to video (b) Link to video

Figure 6: Vehicle getting stuck/not stuck in local minima depending on the horizon. Image a with a horizon of 5 and image b of 30.