Self Driving Car Nano Degree

Model Predictive Control Project

Bastian Dittmar

November 4, 2017

Introduction

In this project, a vehicle driven along a set of waypoints using model predictive control. I polynomial is fit against the next waypoints and then being used to predict the vehicles state at certain time steps. The state update function is based on a kinematic model of the vehicles motion. The minimization of a cost function then leads to actuators (throttle and steering) that achieve the desired vehicle state.

Model

To solve this project a kinematic model of the vehicle is used. The vehicle's state vector has 6 components:

1. vehicle position x
2. vehicle position y
3. vehicle angle ψ towards the x-axis
4. velocity v
5. cross track error cte, the distance to the reference path
6. orientation error $e\psi$, the angular difference to the desired angle $\psi - \psi des$

The state update function for this vector from one time step t to the next time step t_{t+1} with time step length dt, is defined by:

1. $x_{t+1} = x_t + v_t * \cos\psi * dt$
2. $y_{t+1} = y_t + v_t * \sin\psi * dt$
3. $\psi_{t+1} = \psi_t - \frac{v_t}{L_f} * \delta_t * dt$
4. $v_{t+1} = v_t * a_t * dt$
5. $cte_{t+1} = f(x_t) - y_t + v_t * \sin e\psi * dt$
6. $e\psi_{t+1} = \psi_t - \psi des_t - \frac{v_t}{L_f} * \delta_t * dt$

a_t describes the vehicle's acceleration at time step t and δ_t is the steering angle. Both parameters are direct actuators. The goal of model predictive control is to determine optimal values for a and δ at each time step. The model above defines a set of constraints for the state over a number of time steps- Additionally a cost function is is minimized to find the optimal actuators. The cost function is a sum of the following components:

1. $K_{cte} * cte_t$
2. $K_{e\psi} * e\psi_t$
3. $K_v * (v_t - v_{ref})$
4. $K_a * a_t$
5. $K_\delta * \delta_t$
6. $K_{da} * (a_t - a_{t-1})$
7. $K_{d\delta} * (\delta_t - \delta_{t-1})$

Parameters

To achieve a safe and efficient behavior of the vehicle, certain parameters have to be tuned. One set of parameters is the duration of the simulation $T = N * dt$ which is indirectly determined by the number of time steps N and the elapsed time between two steps. If T is chosen too short it will not take the future changes of the route into account. This can lead to very sudden changes in actuators. If T is chosen too large, a lot of unnecessary calculation is done which is dangerous for real time simulations. For the solution of this project a predicted horizon of 1 second was chosen using $N = 10$ and $dt = 0.1$. The time steps are small enough to consider smooth movements but large enough to allow efficient calculation.

The second set of parameters which had to be tuned are the constants for the cost function. The following set seems to be working fine:

1. $K_{cte} = 200$
2. $K_{e\psi} = 200$
3. $K_v = 0.1$
4. $K_a = 1$
5. $K_\delta * 1$
6. $K_{da} * 1$
7. $K_{d\delta} * 10$

The reasoning is quite simple. Obviously the vehicle should be as close to the reference track as possible, this the weights for cte and $e\psi$ are the largest. While $e\psi$ does not have an influence on the current cte, it does so for the next time step as the vehicle will inevitably drive into the wrong direction. The weight for the reference speed is the smallest. Even though we want the vehicle to reach this speed when possible, it should also be allowed to slow down in curves. This seems to be working really nicely as demonstrated by the video. The weight for changing steering input is higher than the other remaining weights. This is supposed to stop the vehicle from oscillating too much.

Latency

One requirement of the project is to consider a latency of 100 ms for the input. This was accounted for by the initial state variable. The state is simply predicted into the future using the described update functions with a value of $dt = 0.1$.