reflections

in this project, I implemented MPC in order to drive the car around the track in the simulation.

MPC is basically converting the problem to an optimization problem. according to that, we need to define the equations (the transition functions), the cost function, and the constraints.

the state in the model include the following:
px = the x position of the car - basically 0 because I transformed it to the car coordinates (in practice - the latency factor add some value to it).
py = the y position of the car - transform to the car coordinates.
psi = the orientation of the car - basically 0 because I transformed it to the car coordinates (in practice - the latency factor add some value to it).
v = the velocity of the car.
cte = the cross-track error of the car.
epsi = the error in the orientation of the car with respect to the desired orientation.

the actuators include:
delta = the steering angle of the car.
a = the acceleration factor. -1 represents full break, and +1 represent full acceleration.

In order to deal with the latency, I just updated the state components as they were supposed to be latency-time into future.

hyperparameters: N and dt: I tried to set N = 25, dt = 0.05, but the results were worse since the complexity is quite big.

in principle, the smaller the dt the better the resolution of the actuators - meaning they are updated every less time so the driving will be smoother.

too small N will cause the model to not taking near-future considerations, like turn after short time. too large N will be redundant because the road will probably change enough so the previous planning will be irrelevant. also, it will be computationally ineffective.