Model Predictive Control (MPC)

Compilation

CRITERIA	MEETS SPECIFICATIONS	STUDENT COMMENTS
Your code should compile.	Code must compile without errors with `cmake` and `make`. Given that we've made CMakeLists.txt as general as possible, it's recommend that you do not change it unless you can guarantee that your changes will still compile on any platform.	The code compiles and the executable(mpc) is created.

Implementation

CRITERIA	MEETS SPECIFICATIONS	STUDENT COMMENTS
The Model	Student describes their model in detail. This includes the state, actuators and update equations.	The model is based on the kinematic m odel discussed in the lesson. Model $x_{t+1} = x_t + v_t * cos(\psi_t) * dt$ $y_{t+1} = y_t + v_t * sin(\psi_t) * dt$ $\psi_{t+1} = \psi_t + \frac{v_t}{L_f} * \delta_t * dt$ $v_{t+1} = v_t + a_t * dt$ $cte_{t+1} = f(x_t) - y_t + v_t * sin(e\psi_t) * dt$ $e\psi_{t+1} = \psi_t - \psi des_t + \frac{v_t}{L_f} * \delta_t * dt$ The state variables are initialized between lines 44-51 of MPC.cpp The parameters are defined to minimize the impact of actuators in lines 12-21. The diff_factors are used to smooth-en the path and the current_factors are used for instantaneous speed and accelerations. The update equations are directly taken from the model equations and are implemented in lines 135-

CRITERIA	MEETS SPECIFICATIONS	STUDENT COMMENTS
		141.
Timestep Length and Elapsed Duration (N & dt)	Student discusses the reasoning behind the chosen *N* (timestep length) and *dt* (elapsed duration between timesteps) values. Additionally the student details the previous values tried.	The values of N and dt are started with 25 and 0.05 as with the values in the lesson. And also the ref_v value is started at 40 as with lesson. But in real life the maximum reference velocity of 40 is unrealistic. Effort is put to increase rev_v value from 40 to 230 gradually before finally settling at 180. In each of these changes the values of N and dt has significant impact in vehicle stability. The observations are 1. When N is increased a little the speed of the vehicle increased. But when in is set to a very high value like 150 or 200 the vehicle literally stopped sometimes as we are looking too much in the future and a slight impact anywhere along the path slows down the vehicle. 2. N and dt also have a relation. If we increase N and decrease dt, I didn’t observe much difference in speed or vehicle behavior. But increasing N and dt has additive impact. 3. Increasing ref_v cause vehicle to go fast and eventualy unstable. Along with N and dt we need to tune other parameters as well. 4. When dt was around 0.1 the car was touching the

CRITERIA	MEETS SPECIFICATIONS	STUDENT COMMENTS
		yellow lines even at lower speeds. When dt was lowers then 0.01 at different speeds, the car was displaying snaky behavior in subsequent laps even though the first lap was good. This is due to the fact we are updating the values too fast. So a balance was needed. And I tried different values between 0.005 to 0.5 before settling in at 0.025.
Polynomial Fitting and MPC Preprocessing	A polynomial is fitted to waypoints. If the student preprocesses waypoints, the vehicle state, and/or actuators prior to the MPC procedure it is described.	The way-points are pre-processed by transforming to car coordinate system in lines 104-111 in main.cpp. A polynomial of 3rd degree is fitted to the transformed way-points. The polynomial coefficients are then used to calculate cte and epsi, which are in turn used to create the trajectory.
Model Predictive Control with Latency	The student implements Model Predictive Control that handles a 100 millisecond latency. Student provides details on how they deal with latency.	A latency of 100 ms is incorporated and is implemented in lines 134-143 in main.cpp.

Simulation

CRITERIA	MEETS SPECIFICATIONS	STUDENT COMMENTS
The vehicle must successfully drive a lap around the track.	No tire may leave the drivable portion of the track surface. The car may not pop up onto ledges or roll over any surfaces that would otherwise be considered unsafe (if humans were in the vehicle). The car can't go over the curb, but, driving on the lines before the curb is ok.	The simulation is recorded in two videos mpc-laps.mp4 and mpc-laps-fast.mp4. The first video(mpc-laps.mp4) is recorded with parameter cost_current_a_factor = 80 in MPC.cpp line 18. The second video(mpc-laps-fast.mp4) is recorded with parameter cost_current_a_factor = 48 in MPC.cpp line 18. These are stored in folder videos in the zip file. The car is pretty much going at the center of the lap in the video mpc-laps.mp4. But the speed is capped at about 50 mph. But I wanted to experiment with speed and stability by varying different parameters. Every time I increased N, dt, and rev_v, and reduced cost_current_a_factor the speed has increased and the stability decreased. The tuning of these parameters for each combination has ensured stability even at higher speeds.