UDACITY

Model Predictive Control (MPC)

A part of the Self-Driving Car Engineer Program

PROJECT REVIEW
CODE REVIEW
NOTES

SHARE YOUR ACCOMPLISHMENT! 🐦 📘

Requires Changes

2 SPECIFICATIONS REQUIRE CHANGES

Shiny Student!
The solution was almost completed🎉
A great job has been done in the project and i must admit that i enjoyed reviewing the project. The implementation is great👍 Please i have provided some few comments and amendment to be looked at. I can't wait to see a wonderful resubmission, keep learning and stay Udacious!!
To expand ones knowledge in this area, it might be good in one's extra time to look at the following materials:

- MPC basics with code examples
- Further understand MPC from this resource.
- A Gentle Introduction to Model Predictive Control (MPC) Formulations based on Discrete Linear State Space Models
- Good tutorial on MPC here too

Compilation

✓

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.

Nice job in this section of the project, the project compiles without any errors with `cmake` and `make` 👍. Keep it up!

```
    for (int i = a_start; i < n_vars; i++) {
                         ^
/home/kenneth/Downloads/190/tommytracey-Udacity-CarND-Term2-da71bea/p5-model-pre
dictive-control/src/MPC.cpp:200:21: warning: comparison between signed and unsig
ned integer expressions [-Wsign-compare]
    for (int i = 0; i < n_constraints; i++) {
                      ^
/home/kenneth/Downloads/190/tommytracey-Udacity-CarND-Term2-da71bea/p5-model-pre
dictive-control/src/MPC.cpp:263:21: warning: comparison between signed and unsig
ned integer expressions [-Wsign-compare]
    for (int i = 0; i < N-1; i++) {

[ 66%] Building CXX object CMakeFiles/mpc.dir/src/main.cpp.o
/home/kenneth/Downloads/190/tommytracey-Udacity-CarND-Term2-da71bea/p5-model-pre
dictive-control/src/main.cpp: In lambda function:
/home/kenneth/Downloads/190/tommytracey-Udacity-CarND-Term2-da71bea/p5-model-pre
dictive-control/src/main.cpp:149:29: warning: comparison between signed and unsi
gned integer expressions [-Wsign-compare]
          for (int i = 2; i < vars.size(); i ++) {
                            ^
[100%] Linking CXX executable mpc
[100%] Built target mpc
```

Implementation

✓

Student describes their model in detail. This includes the state, actuators and update equations.

Great job done in describing the state, actuators and updates equations in the file `readme.md` as required👏

✓

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 discussion on the choice of `N` and `dt` was just awesome and it was mentioned in the readme file that quite a number of values were tried for `N` and `dt` before finally settling on the chosen ones which is really great. Kudos!

🔄

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.

A lot of effort has been put in this project👍

Required

This question requires that description on how the points were preprocess and fitted into the polynomial should be included in the reflection file.

Suggestions

- Please, it's also important to reference the section where the implementation was done in readme file.
- Please, information on how the waypoints were converted should be provided in the writeup file,

🔄

The student implements Model Predictive Control that handles a 100 millisecond latency. Student provides details on how they deal with latency.

A nice job has been done in the project implementation, some issues about latency are still to be addressed in a writeup file.

Required

This question requires that information on how latency was handle should be provided in the documentation file.

Suggestions

Please, I can see how this was clearly implemented in `main.cpp`, provide discussion in the writeup file as required by this rubric.
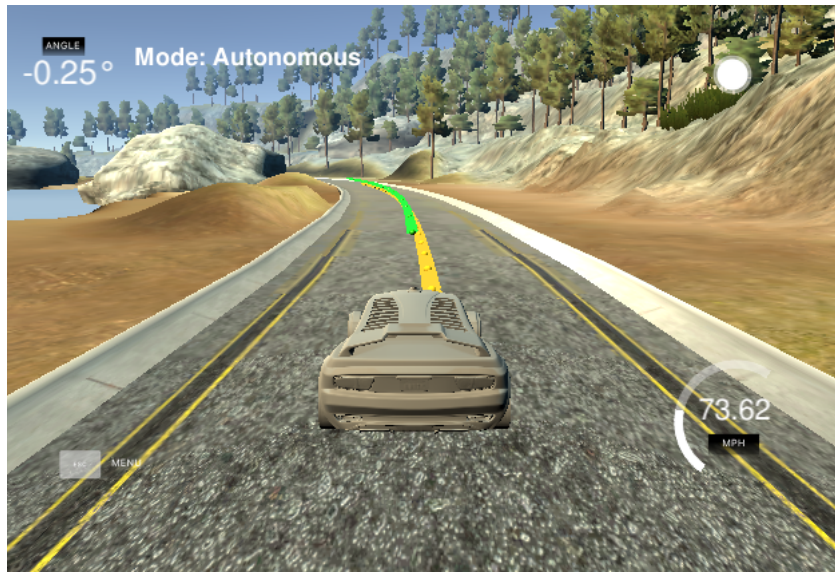
Simulation

✓

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 was stable and completed the track without going out track. Keep it up!

RESUBMIT

DOWNLOAD PROJECT

Best practices for your project resubmission

Ben shares 5 helpful tips to get you through revising and resubmitting your project.

⊙ Watch Video (3:01)

RETURN TO PATH