CarND-Controls-MPC

Self-Driving Car Engineer Nanodegree Program

Reflection

The Model

Kinematic model is introduced in this project.

This model includes following states: vehicle's longitudinal(x), lateral (y), oritentation (psi), velocity(v), cross track error(cte) and psi error (epsi).
The acutator includes steering (δ) and acceleartion(a) (incl. decelleration).
The model use actuator as inputs and update the states according to following equations:

- $x(t+1)=x(t)+v(t)*\cos(psi(t))*dt$
- $y(t+1)=y(t)+v(t)*\sin(psi(t))*dt$
- $psi(t+1)=psi(t)+Lf/v(t)*\delta(t)*dt$
- $v(t+1)=v(t)+a(t)*dt$
- $cte(t+1)=f(x(t))-y(t)+(v(t)*\sin(epsi(t))*dt)$
- $epsi(t+1)=psi(t)-psides(t)+(Lf/v(t)*\delta(t)*dt)$

where: Lf is the distance from front axle to vehicle center of gravtiy, psides is target psi.

Timestep Length and Elapsed Duration (N & dt)

In my implementation, timestep length N is `20`, dt is set to `0.1`. The combination of these two values means the controller will look ahead `2` seconds for the optimazation. smaller dt or biggher N lead to more computation resources. Too big dt or small N means short distance ahead is considered. If vehicle in high speed it is not sufficient for a optimize control.

I also tried the combinaton of `N=20`, dt `=0.05`.

Polynomial Fitting and MPC Preprocessing

The waypoints were transformed to vehicle's coordinate system. After tranformation, the vehicle's x, y and the orientation angle are `0`s. This processing simplified the fitting of polynomial to the waypoints. Also vehicle speed is transformed from mph to m/s.

Model Predictive Control with Latency

The latency of vehicle actuators are handled in this way: `100`ms actuation delay are introudced to the actuators (line `99-103` in MPC.cpp). MPC controller take this latency into account and handled well after optimization.

Dependencies

- cmake >= 3.5
- All OSes: click here for installation instructions
- make >= 4.1(mac, linux), 3.81(Windows)
- Linux: make is installed by default on most Linux distros
- Mac: install Xcode command line tools to get make
- Windows: Click here for installation instructions
- gcc/g++ >= 5.4
- Linux: gcc / g++ is installed by default on most Linux distros
- Mac: same deal as make - [install Xcode command line tools]((https://developer.apple.com/xcode/features/)
- Windows: recommend using MinGW
- uWebSockets
- Run either `install-mac.sh` or `install-ubuntu.sh`.

- If you install from source, checkout to commit `e94b6e1`, i.e. `git clone https://github.com/uWebSockets/uWebSockets cd uWebSockets git checkout e94b6e1` Some function signatures have changed in v0.14.x. See this PR for more details.

- **Ipopt and CppAD:** Please refer to this document for installation instructions.

- Eigen. This is already part of the repo so you shouldn't have to worry about it.
- Simulator. You can download these from the releases tab.
- Not a dependency but read the DATA.md for a description of the data sent back from the simulator.

Basic Build Instructions

1. Clone this repo.
2. Make a build directory: `mkdir build && cd build`
3. Compile: `cmake .. && make`
4. Run it: `./mpc`.

Tips

1. It's recommended to test the MPC on basic examples to see if your implementation behaves as desired. One possible example is the vehicle starting offset of a straight line (reference). If the MPC implementation is correct, after some number of timesteps (not too many) it should find and track the reference line.
2. The `lake_track_waypoints.csv` file has the waypoints of the lake track. You could use this to fit polynomials and points and see of how well your model tracks curve. NOTE: This file might be not completely in sync with the simulator so your solution should NOT depend on it.

3. For visualization this C++ matplotlib wrapper could be helpful.)
4. Tips for setting up your environment are available here
5. **VM Latency:** Some students have reported differences in behavior using VM's ostensibly a result of latency. Please let us know if issues arise as a result of a VM environment.

Editor Settings

We've purposefully kept editor configuration files out of this repo in order to keep it as simple and environment agnostic as possible. However, we recommend using the following settings:

- indent using spaces
- set tab width to 2 spaces (keeps the matrices in source code aligned)

Code Style

Please (do your best to) stick to Google's C++ style guide.

Project Instructions and Rubric

Note: regardless of the changes you make, your project must be buildable using cmake and make!

More information is only accessible by people who are already enrolled in Term 2 of CarND. If you are enrolled, see the project page for instructions and the project rubric.

Hints!

- You don't have to follow this directory structure, but if you do, your work will span all of the .cpp files here. Keep an eye out for TODOs.

Call for IDE Profiles Pull Requests

Help your fellow students!

We decided to create Makefiles with cmake to keep this project as platform agnostic as possible. Similarly, we omitted IDE profiles in order to we ensure that students don't feel pressured to use one IDE or another.

However! I'd love to help people get up and running with their IDEs of choice. If you've created a profile for an IDE that you think other students would appreciate, we'd love to have you add the requisite profile files and instructions to ide_profiles/. For example if you wanted to add a VS Code profile, you'd add:

- /ide_profiles/vscode/.vscode
- /ide_profiles/vscode/README.md

The README should explain what the profile does, how to take advantage of it,

and how to install it.

Frankly, I've never been involved in a project with multiple IDE profiles before. I believe the best way to handle this would be to keep them out of the repo root to avoid clutter. My expectation is that most profiles will include instructions to copy files to a new location to get picked up by the IDE, but that's just a guess.

One last note here: regardless of the IDE used, every submitted project must still be compilable with cmake and make./

How to write a README

A well written README file can enhance your project and portfolio. Develop your abilities to create professional README files by completing this free course.