

Model Predictive Control notes

Philippe Weingertner

August 10, 2018

1 Motion Planning problem

Formulation of the problem.

2 PID control recap

2.1 Proportional Control

2.2 Oscillations and PD Controller

2.3 Systematic Bias and PID Controller

2.4 Twiddle and parameter optimization

3 Vehicle Models

3.1 Dynamic vs Kinematic Models

3.2 Kinematic Model

3.2.1 State

3.2.2 Global Kinematic Model

3.2.3 Following Trajectories

3.2.4 Fitting Polynomials

3.2.5 Errors

3.3 Dynamic Models

Forces, Slip Angle, Slip ratio and Tire Models

3.4 Actuator Constraints

4 Model Predictive Control

MPC reframes the task of following a trajectory as an optimization problem. The solution to the optimization problem is the optimal trajectory.

MPC involves simulating different actuator inputs, predicting the resulting trajectory and minimizing a set of constraints (or cost functions).

Input: a reference trajectory we want to follow

Constraints:

- Vehicle Model
- Comfort

Output: actuator commands (steering, throttling, braking ...)

Once we found the lowest cost trajectory, we implement the very first set of actuation commands. Then we throw away the rest of the trajectory we calculated. Instead of using the old trajectory we predicted, we take our new state and use that to calculate a new optimal trajectory. In that sense, we are constantly calculating inputs over a future horizon. That's why this approach is also called Receding Horizon Control. We constantly reevaluate the trajectory because our vehicle model is not perfect and the next predicted (or planned) state may (slightly...) differ with our prediction (in the sense of a consequence of a command sent).

4.1 Optimization under constraints

4.2 Defining cost functions

4.3 Length and duration

4.4 MPC Solver algorithm

4.5 Latency handling

A contributing factor to latency is actuator dynamics. For example the time elapsed between when you command a steering angle to when that angle is

actually achieved. This could easily be modeled by a simple dynamic system and incorporated into the vehicle model. One approach would be running a simulation using the vehicle model starting from the current state for the duration of the latency. The resulting state from the simulation is the new initial state for MPC.

Thus, MPC can deal with latency much more effectively, by explicitly taking it into account, than a PID controller.

4.6 MPC Solver implementation with ipopt and cppAD