

# MPC Project

## The Model

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

The car dynamics is modelled with the bicycle model. The car travels tangential to a heading direction. The velocity is changed by an acceleration control value. The heading is changed by a steering angle. The exact rate of change of the heading is also impacted by the distance between the front and back wheels in the model and the current speed.

So there are four state variables, and two control variables. The state variables are two coordinates,  $x$  and  $y$ , the velocity,  $v$ , and the heading  $\psi$ . The control variables are the acceleration,  $a$ , and the steering angle  $\delta$

The state is then updated each time point,  $t_i$ , according to the following equations:

$$\begin{aligned}x(t_{i+1}) &= x(t_i) + v \cos(\psi(t_i))\Delta t \\ y(t_{i+1}) &= y(t_i) + v \sin(\psi(t_i))\Delta t \\ \psi(t_{i+1}) &= \psi(t_i) + v\Delta t\delta(t_i)/L_f \\ v(t_{i+1}) &= v(t_i) + a(t_i)\Delta t\end{aligned}$$

where  $\Delta t$  is the timestep, i.e.,  $t_{i+1} = t_i + \Delta t$ .

## Timestep size and prediction length

*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 timestep impact both the accuracy of the predictions and the latency, as a small timestep typically implies a lot of calculation, which takes time.

In practice, our final trajectories are very smooth, so accuracy between timesteps is not a huge issue. Moreover given that there is a latency of at least 100ms, there is not much point having a time step much smaller than that because any controls calculated in that time period would be out of date before they could be used. So I chose a value just below this threshold, 50ms.

For the size of N, the number of time steps, my main concern was having enough points to capture the important features of the upcoming road. Too few points and you may as well plot a straight trajectory. On the other hand, too many points leads to excessive calculations too.

Adding too many points can make the MPC become unstable too. This is because as predict into the future, uncertainty increases about what your state will be, but those points are contributing the same cost to points nearer to now, so this may bias the prediction, particularly in the case of a noisy measurement of the current state. This could be avoided by discounting the cost of points as they go further into the future.

## 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.*

### Waypoint curve

I fitted a cubic polynomial to the waypoint. My reasoning here was that it is the simplest curve that can fit all the shapes on the test track, the most complicated shape being the s-shape between two turns.

### Car-centered Frame of Reference

The waypoints were first all transformed into a car centered frame of reference where the car is travelling in the positive  $x$  direction. This has the nice feature that the initial state of the car is quite simple, 0, 0, 0,  $v$ . Also, as long as the curve of the road does not curve too much, its shape can be described in the form  $y = f(x)$ , where we fit a function  $f$  from the way points. If this was not true, i.e., if the waypoints bent right back on themselves in the car frame, then the waypoints would have to be parametrised with two functions, i.e.,  $x = f(t)$  and  $y = h(t)$

### Acceleration Control

I before the MPC step, I infer the acceleration from subsequent velocity measurements. I did this because the telemetry does not give anything that reliably tells us this directly. In some cases the throttle acts like acceleration, however this is deceptive. Firstly, because the car in the simulation has rolling resistance, a non-zero throttle is required to maintain a constant speed, whereas acceleration in this case is zero. Similarly a car rolling with zero throttle will slow down, i.e., have a small negative acceleration. Secondly, the throttle is bounded, but acceleration has no inherent limits. N.B., in my MPC implementation I did constrain acceleration. This is to avoid the throttle control becoming saturated, (i.e., hitting its bounds.)

After the MPC, I use a PID controller to get a value of the throttle control from the desired acceleration value.

## 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.*

Objective function includes:

- The cost associated with a trajectory that does not follow the waypoint curve. (The cross track error.)
- The cost associated with a trajectory that is not tangential to the of the waypoint curve.
- Calculate the cost associated with a trajectory that does not keep up with the speed set point.
- The cost associated a control trajectory that changes rapidly. This attempts to smooth the controls by minimising the difference between successive control values.
- The cost associated with a trajectory that corners too hard. The cost per time step here is proportional to the centripetal force.

The constraints I used were:

- A constraint equation for each time step of the state variable to ensure they obey the model equations from the first section. These are equality constraints.
- A constraint on the state variables of the first timestep to ensure they equal the current measured state.
- A constraint on the control variable at all times to keep them in a fixed range. For the steering this makes sense because they have a hard physical limit in the car design. The acceleration is also similarly limited by the maximum throttle.

To deal with the latency, I take the control value that the MPC gives at the time equal to the total latency. This I took to be 3 time steps into the prediction, or 150ms. This is roughly the artificial latency (100ms) plus the round trip transmission time and the time it took to do the calculations. N.B., this means that the result is a bit dependent on the system it runs on, beacuse the calculation and transmission times will vary.