Top group to look at in this space: Chris Gerdes’ group @ Stanford

They developed a lot of different models to compare fidelity and level of accuracy need in control.

Look at papers and PhD presentations from that lab.

Look at John K. Subosits dissertation.

“Impacts of Model Fidelity on Trajectory Optimization for Autonomous Vehicles in Extreme Maneuvers”

For simulation, you want the highest fidelity possible, but for MPC it might be better to simplify.

Find the simplest model you can get away with!

A dumber model at a higher rate is often better than the more complex model at a lower Hz.

In MPPI, they use even dumber models (sometimes learned). Very “hand-wavy” way to wrap model error into some stochasticity.

For off-road, we’re not near where humans are (humans don’t do the double track thing in our heads).

Zac recommends going to a dynamic single track for developing the controller, then use the double track to test the controller on the environment.

Wrap the controller based on dynamic single track in the double track model. The model fidelity of the single track should be enough to generate the controls, then the double track will better simulate the environment on that generated control.

Point mass wouldn’t really work here (since point mass is just forces and cars are over constrained).

The kinematic single track model captures steering and kinematics in that it can’t immediately translate sideways and has to turn based on steering.

The dynamic single track treats it as a rigid body and captures throttle and steering.

Basically, all of the car companies do iLQR on single-track model.

This kind of becomes a qualitative art in developing the cost functions and coming up with the simplest model we can get away with.

MPPI and MPC will have similar model-based issues.

Stiff ODEs is one way of considering ill-conditioning, there’s stuff in the dynamics that makes the A matrix have a huge spread in the eigenvalues (i.e. some dynamics that are very fast [“stiff springs”] and some are very slow). Typically, these dynamics should be separated (i.e. quadrotors have a separate position controller and attitude controller because one is much faster, combining them would be rough). Some dynamics only have to run at ~10Hz (car stability) but some need 100s of Hz (terrain analysis).

Most real-world implementation have multiple controllers; we probably need as well. Ex. In this space, in a real car, there needs to be a controller that turns the steering wheel with some ~kHz rate with some PD controller.

Have one person write up the single-track model and test the MPC on it.

Have one person work on the ground truth simulation with the double track model on the terrain.

For MPPI, build on the single-track model as well.

MPC on single-track:

1st step: code up model. John Subosits papers are good. [Find John’s dissertation, it will have everything in one place] He did a ton of work on MPC, mostly on the single-track model.

“Long-Horizon Vehicle Motion Planning and Control Through Serially Cascaded Model Complexity” – Used single track model for the first part of the horizon, then the point mass model for the rest of the MPC horizon so the controller can run at much faster Hz.

Pick your favorite algorithm for MPC, great one is OSQP (which we did!).