Problem and Motivation

Model Predictive Control (MPC) is a technique that some are investigating for control of autonomous vehicles. MPC uses a model of the system to predict its behavior and plans trajectories that achieve some properties. At every point in time, MPC plans a sequence of control inputs over some prediction horizon and executes the first input.

It can be difficult to understand the controller’s behavior from looking at the vehicle states alone, which is already a lot of data to view. We additionally have the planned trajectories at every point in time, which is very difficult to analyze in a static image.

Goal: minimize lateral error, heading error, and steering changes, while prioritizing obstacle/road bound avoidance

\[
\min_{u,x} \sum_{k=1}^{30} \left( \|x^k\|_{Q_e^k} + \|\Delta u^k\|_{R^k} + \|\psi - \psi_r\|_{Q_{\psi}^k} + \sum_{k=11}^{30} \|\sigma_{obs}^k\|_{W_{obs}^k} \right)
\]

subject to:

vehicle model: \[ x^{k+1} = A^k x^k + B_1^k u^k + B_2^k u^{k+1} + C^k \]

obstacle avoidance: \[ H_{obs}^k x^k \leq G_{obs}^k + \sigma_{obs}^k \]

Approach

States, inputs, and time are qualitative variables, so they are encoded by position in the form of line charts. The planned states and inputs are also qualitative variables (and encoded) as position, but they also represent the terms in the cost function, which are nominal variables. To account for this, they are additionally encoded with hue.

This project uses Bokeh, a Python library for creating interactive visualizations in a web browser. Analysis and prototyping new calculations can be easily done in Python on the server side and updates passed to browser clients.

Results

The resulting visualization shows time traces of relevant signals, and plots the position of the vehicle. A slider at the top allows a user to scrub forward or backward in time, and the planned trajectories at that point in time are displayed. Additionally, a chart at the bottom displays, for each point in the predicted trajectory, the contribution for each term in the cost function. This can help designers understand the controller’s behavior.

Future Work

Bokeh is well set up to support streaming data, so one possible extension would be to allow visualization of the controller’s planned trajectories in real time as it calculates them. This might be useful not only as an engineering and development tool, but also for consumer comfort and acceptance.

Additionally, real (experimental) data is noisy. Handling how to display the noise in a meaningful way would be an interesting extension to this project.