Optimization of PID controller parameters for simulation of autonomous vehicle path-tracking

Design Strategy

Several approaches were considered during the first week of the project, and it was decided that the project should be split into three parts, first fitting an objective function, then testing and validating the function, and finally optimizing this function.

As we have access to a dataset containing the simulation results for 78 different controller gains, we decided that we should train and validate a model that estimates the lateral error and the longitudinal speed error. Once we have a validated model, we will have an accurate and robust objective function to optimize using surrogate model optimization.

Our plan is to approach the problem using the methodology described in the textbook for fitting a noisy objective function using Gaussian radial basis functions. For this project we will be using Mean Squared Error as our metric for evaluating the generalization error. We will use cross-validation to estimate the generalization error for a range of smoothing parameters λ. Once the optimal smoothing parameter is determined we would have achieved the most robust objective function for the problem.

We will then use a series of built-in optimization libraries in the SciPy package to find an optimal solution X^*. This would then also allow us to evaluate the performance of several algorithms and determine the most effective solver for the problem.

Work Completed

Our first step was formatting the dataset, splitting the dataset into the 4 distinct simulations, and highlighting the design points that successfully fell within the lateral error and speed error constraint.

A python script was developed that imports the data from our excel spreadsheet and creates:

- a list of design points $X \in \mathbb{R}^3$, containing the three PID gains K_i, K_p, and K_d
- a list of respective output values $Y \in \mathbb{R}^2$, containing the lateral errors and speed errors

Timeline for project

May 18th: Completed setup code (All packages imported and all subfunctions written)
May 21st: Implement at least one of the SciPy optimization algorithms
May 22nd: Two to three more optimization algorithms are evaluated and compared
May 24th: All plotting is complete and ready for LaTeX submission
May 29th: LaTeX documentation is ready for submission