# 1 Introduction

# 2 Problem Formulation

## 2.1 Control System

For our simulations, we are modelling the dynamics of the car with a simple non-linear bicycle model, with the following state ($x$) and control ($u$)

$$x = \begin{bmatrix} p_x & p_y & \theta & \delta & v \end{bmatrix}^\top \qquad u = \begin{bmatrix} a & \dot{\delta} \end{bmatrix}^\top$$

where $p_x, p_y$ is the position, $\theta$ is the orientation, $\delta$ is the steering angle, and $v$ is the velocity. The controls for the bike are acceleration $a$, and steering angle rate $\dot{\delta}$.

In order to make the system control-affine for easy optimization, we linearize our approximate dynamics model about $X_{ref}$ and $U_{ref}$ to get the following Jacobians for each step $k \in [1, N]$ in the trajectory:

$$A_k = \left.\frac{\partial f}{\partial x}\right|_{x_{ref,k}, u_{ref,k}} \qquad B_k = \left.\frac{\partial f}{\partial u}\right|_{x_{ref,k}, u_{ref,k}}$$

where $f(x, u)$ is our approximate discrete dynamics model, $X_{ref}$ is the optimal trajectory computed offline with approximate dynamics model, and $U_{ref}$ is the optimal controls computed offline with approximate dynamics model.

With this, the system becomes control-affine, where the states are given by

$$x_{k+1} = A_k x_k + B_k u_k$$

and the proportional-derivative (PD) controller gives the control

$$u_k = -\left[P \cdot (x_k - x_{ref,k}) + D(x_k - x_{k-1})\right]$$

where $P, D$ are the proportional and derivative control matrices to be found by our algorithm.

## 2.2 Optimizing for PD matrices

### 2.2.1 Without Regularization

In order to automatically tune the PD matrices, we optimize for best PD matrices that minimize the quadratic cost of the trajectory generated by the PID controller, as constrained by control-affine system dynamics. Mathematically, we attempt to solve the following problem:

$$\begin{aligned}
\min_{P,D} \quad & \left(X - X_{ref}\right)^\top Q \left(X - X_{ref}\right) \\
\text{s.t.} \quad & x_1 = x_{ref,1} \\
& x_{k+1} = A_k x_k + B_k u_k \\
& u_k = -\left[P \cdot (x_k - x_{ref,k}) + D(x_k - x_{k-1})\right]
\end{aligned}$$

where $X, X_{ref}$ are the generated and reference trajectories respectively (a direct function of the states $x_k$), $Q \succeq 0$ is a quadratic cost matrix, and $A_k, B_k$ are the Jacobians representing the linearized system dynamics as mentioned in Section 2.1 above.

We note that the cost function is quadratic and the constraints are affine, thus we have a quadratic programming problem, which can be solved easily.

## 2.3 With Regularization

As noted in Section 3.1.1, the simple formulation results in exploding P and D values. To combat this, we add L1 regularization to the quadratic cost as follows:

$$\begin{aligned}
\min_{P,D} \quad & \left(X - X_{ref}\right)^\top Q \left(X - X_{ref}\right) + \lambda \left( \|P\|_1 + \|D\|_1 \right) \\
\text{s.t.} \quad & x_1 = x_{ref,1} \\
& x_{k+1} = A_k x_k + B_k u_k \\
& u_k = -\left[P \cdot (x_k - x_{ref,k}) + D(x_k - x_{k-1})\right]
\end{aligned}$$

where $\lambda$ is a hyperparameter that affects the amount of regularization in the system.

Although the L1 regularization makes the problem no longer quadratic, the problem is still convex and can be solved efficiently by proximal gradient descent.

# 3 Results

## 3.1 Single trajectory

### 3.1.1 Without Regularization

$$P = \begin{bmatrix} -25.48 & -43.88 & 228.95 & 904.98 & 1151.69 \\ -1.81 & -4.19 & 29.26 & -70.38 & -26.2 \end{bmatrix}$$

$$D = \begin{bmatrix} -130.93 & -71.37 & -1142.36 & 22.02 & 2424.28 \\ -9.39 & 10.48 & -19.01 & 3.23 & 186.85 \end{bmatrix}$$

### 3.1.2 With Regularization

$$P = \begin{bmatrix} -0.79 & 0.11 & 0 & 0 & 0 \\ -0.22 & -0.08 & 0 & 0 & 0 \end{bmatrix}$$

$$D = \begin{bmatrix} -0.42 & 0 & 0 & 0 & 0 \\ 0.2 & 0 & 0 & 0 & 0 \end{bmatrix}$$