Model Predictive Control for Trajectory Following with Actuator Failure

Thomas Lipp
EE364b: Convex Optimization II Class Project

Introduction

Desirable to have automated vehicles follow a given trajectory
Traditional approaches:
  - Value function solution
  - Link trajectory and desired actuator inputs at trajectory creation
  - Cannot adapt to changes in actuator output (degradation or failure)
  - Model predictive control (MPC) can account for actuator changes

The Trajectory Following Problem

- Discrete vehicle dynamics:
  \[ x_{t+1} = F(x_t) + G(x_t, u_t) \tag{1} \]
  - Vehicle state and control inputs \( x_t \in \mathbb{R}^m, u_t \in \mathbb{R}^m \)
  - Vehicle dynamics \( F : \mathbb{R}^m \to \mathbb{R}^m \)
  - Control dynamics \( G : \mathbb{R}^m \times \mathbb{R}^m \to \mathbb{R}^m \)
  - Desired trajectory is \( x_t \in \mathbb{R}^m, t = 0, \ldots, T \)
  - The trajectory following problem can be stated as:
    \[
    \begin{align*}
    \text{minimize} & \quad \sum_{t=0}^{T-1} (L(u_t) + D(x_{t+1}, x_t+1)) \\
    \text{subject to} & \quad x_{t+1} = F(x_t) + G(x_t, u_t), \quad t = 0, \ldots, T-1 \\
    & \quad u_t \in U, \quad t = 0, \ldots, T-1 \\
    & \quad x_t \in X, \quad t = 1, \ldots, T
    \end{align*}
    \tag{2} \]

- variables:
  - Vehicle state \( x_t, t = 0, \ldots, T \)
  - Control inputs \( u_t, t = 0, \ldots, T-1 \)

- problem data:
  - Time \( T \), initial state \( x_{\text{initial}} \)
  - Trajectory \( x_t, t = 0, \ldots, T \)
  - Vehicle dynamics \( F(x_t, G(x_t, u_t) \)
  - Control law \( L : \mathbb{R}^m \times \mathbb{R}^m \to \mathbb{R}^m \)
  - Deviation from trajectory cost \( D : \mathbb{R}^m \times \mathbb{R}^m \to \mathbb{R}^m \)
  - Set of valid states and inputs \( X \) and \( U \)

The LQR Approach

- Linear quadratic regulation (LQR) is a well studied approach
- LQR has closed form solutions solved with algebraic Riccati equations
- LQR formulation:
  - Assumes linear vehicle dynamics
  - Takes \( L \) and \( D \) to be quadratic
  - Takes \( X = \mathbb{R}^m, U = \mathbb{R}^m \)
  - No ability to encode actuator constraints

A Model Predictive Control Approach

- Even if tractable, equation (2) may be too large to be solved in real time.
- Instead at time \( t \) with state \( x_{\text{current}} \), look ahead \( s \) time steps, and take action \( u_t \) given by:
  \[
  \begin{align*}
  \text{minimize} & \quad \sum_{t+i=0}^{T-1} (L(u_t) + R(S, x_t)) + V(x_{t+s}, x_{t+1}) \\
  \text{subject to} & \quad x_{t+i} = A_t x_t + B_t u_t + C_t, \quad i = t, \ldots, t + s - 1 \\
  & \quad x_t = x_{\text{current}}, \quad u_t \in U, \quad t = t, \ldots, t + s - 1 \\
  & \quad x_t \in X, \quad t = t, \ldots, t + s 
  \end{align*}
  \tag{3} \]

- problem data:
  - \( L \), \( s \), \( x_{\text{Current}} \)
  - Linearized dynamics about \( \bar{x}_t \): \( A_t \in \mathbb{R}^{m \times m}, B_t \in \mathbb{R}^{m \times m} \), \( \bar{C}_t \in \mathbb{R}^m \)
  - Convex control cost \( L \), for example quadratic
  - \( X \) and \( U \) convex.
  - Encode control constraints on \( U \) such as actuator failure: \( M u_t < N \)
  - Convex regularization cost \( \bar{R} \) and final deviation cost \( \bar{V} \)

Example: Race Car Turn Formulation

- car has state \( x = [x_{\text{pos}}, y_{\text{pos}}, y_{\text{vel}}] \)
- control \( u = [u_t, u_y] \)
- race car modeled by discrete dynamics with time step 0.1:
  \[
  A_t = \begin{bmatrix}
  1.0 & 0.1 & 0.0 \\
  0 & 1.0 & 0.0 \\
  0 & 0 & 1.0
  \end{bmatrix}, \quad B_t = \begin{bmatrix}
  0.005 & 0.0 \\
  0.1 & 0.0 \\
  0 & 0.1
  \end{bmatrix}, \quad C_t = \begin{bmatrix}
  0 \\
  0 \\
  0
  \end{bmatrix}
  \forall t \tag{4} \]

Example: Race Car Turn Results

- The designed input path uses the designed inputs from trajectory generation
- Impact of varying \( s \) on the performance of MPC

Example: Race Car, Effects of Look Ahead

- Infeasible trajectory: performance improves when the vehicle is allowed greater freedom (larger \( s \)) in returning to the trajectory
- Feasible trajectory with no \( L \) cost, noise, or over-actuation: greedy control is optimal and recovers the designed inputs

Conclusions

- Including actuator constraints, which traditional approaches do not allow, produces significant performance improvements.
- MPC formulation can be easily adapted to account for nonlinear dynamics, road constraints, and slew limits on control inputs.