MPC based Trajectory Following Controller

1、MPC Controller Design

1.1 System Model

Due to low vehicle speed, we ultilze the kinematic model as the system model:

$$\begin{cases} \dot{p}_x = v\cos(\phi) \\ \dot{p}_y = v\sin(\phi) \\ \dot{\phi} = \dfrac{v}{L}\tan(\delta) \\ \dot{v} = a \end{cases}$$

State: $X = [x\ y\ \phi\ v]^T$, Input: $U = [a\ \delta]$

- $Linearization$

  For improving the solving efficiency, we linearize the nonlinear system model. There are two different method.

  - Linearization along the state trajectory:

    We can linearize the system model based on the result of last run.

  - **Linearization along the reference trajectory**

    we ca linearize the system mode based on the reference state.

Define the linearization point as follows:

$\bar{X} = [\bar{p_x}\ \bar{p_y}\ \bar{phi}\ \bar{v}]^T$, $\bar{U} = [\bar{a}\ \bar{\delta}]^T$

The linear model:

$$\begin{bmatrix} \dot{p_x} \\ \dot{p_y} \\ \dot{\phi} \\ \dot{v} \end{bmatrix} = \begin{pmatrix} 0 & 0 & -\bar{v}\sin\bar{\phi} & \cos\bar{\phi} \\ 0 & 0 & \bar{v}\cos\bar{\phi} & \sin\bar{\phi} \\ 0 & 0 & 0 & \frac{\tan\bar{\delta}}{L} \\ 0 & 0 & 0 & 0 \end{pmatrix} \begin{bmatrix} p_x \\ p_y \\ v \\ \phi \end{bmatrix} + \begin{pmatrix} 0 & 0 \\ 0 & 0 \\ 0 & \frac{\bar{v}}{L}\frac{1}{\cos^2\bar{\delta}} \\ 1 & 0 \end{pmatrix} \begin{bmatrix} a \\ \delta \end{bmatrix} + \begin{pmatrix} \bar{v}\sin\bar{\phi} \\ -\bar{v}\bar{\phi}\cos\bar{\phi} \\ -\bar{v}\frac{\bar{\delta}}{L\cos^2\bar{\delta}} \\ 0 \end{pmatrix}$$

- Discretizaton

  For computer calculation, we discretize the continuous system with forward Euler difference:

$$\begin{aligned} A_{\text{discrete}} &= I + \Delta T \times A_{\text{linearized}} \\ B_{\text{discrete}} &= \Delta T \times B_{\text{linearized}} \\ g_{\text{discrete}} &= \Delta T \times g_{\text{linearized}} \end{aligned}$$

1.2 Define Cost Function

The objective is to follow the trajectory accuracy, the cost function can define as follow:

$$
\begin{aligned}
&\min \sum_{k=1}^{N}(x_k-\bar{x}_k)^T Q(x_k-\bar{x}_k)+u_{k-1}^T R u_{k-1}\\
&\text{s.t. } x_0=x,\\
&x_{k+1}=f\left(x_k,u_k\right), \quad k=0,\ldots,N-1\\
&\underline{x}\le x_k\le \bar{x}, \quad k=1,\ldots,N\\
&\underline{u}\le u_k\le \bar{u}, \quad k=0,\ldots,N-1
\end{aligned}
$$

1.3 Prediction

$$\eta = AAx_0 + BBU + G$$

$\eta = [x_1\ x_2\ \ldots x_N]^T, U = [u_0\ u_1\ u_2 \ldots u_{N-1}]^T$

$$AA = \begin{bmatrix} A_0 \\ A_1A_0 \\ \vdots \\ \prod_{k=0}^{N-1} A_k \end{bmatrix}$$

$$BB = \begin{bmatrix} B_0 & 0 & \ldots & 0 & 0 \\ A_1B_0 & B_1 & \ldots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ \prod_{k=1}^{N-1} A_kB_0 & \prod_{k=2}^{N-1} A_kB_1 & \ldots & A_{N-1}B_{N-2} & B_{N-1} \end{bmatrix}$$

$$G = \begin{bmatrix} g_0 \\ A_1g_0 + g_1 \\ \vdots \\ \sum_{n=0}^{N-2}\left(\prod_{k=n+1}^{N-1} A_k\right)g_n + g_{N-1} \end{bmatrix}$$

Thus we can reomove the equility constrain of the optimize problem.

1.4 OSQP Solver

we adopt the QP solver OSQP to solve the QP problem.

OSQP solves convex quadratic programs (QPs) of the form

$$
\begin{aligned}
\text{minimize} \quad & \frac{1}{2}x^T P x + q^T x \\
\text{subject to} \quad & l \le Ax \le u
\end{aligned}
$$

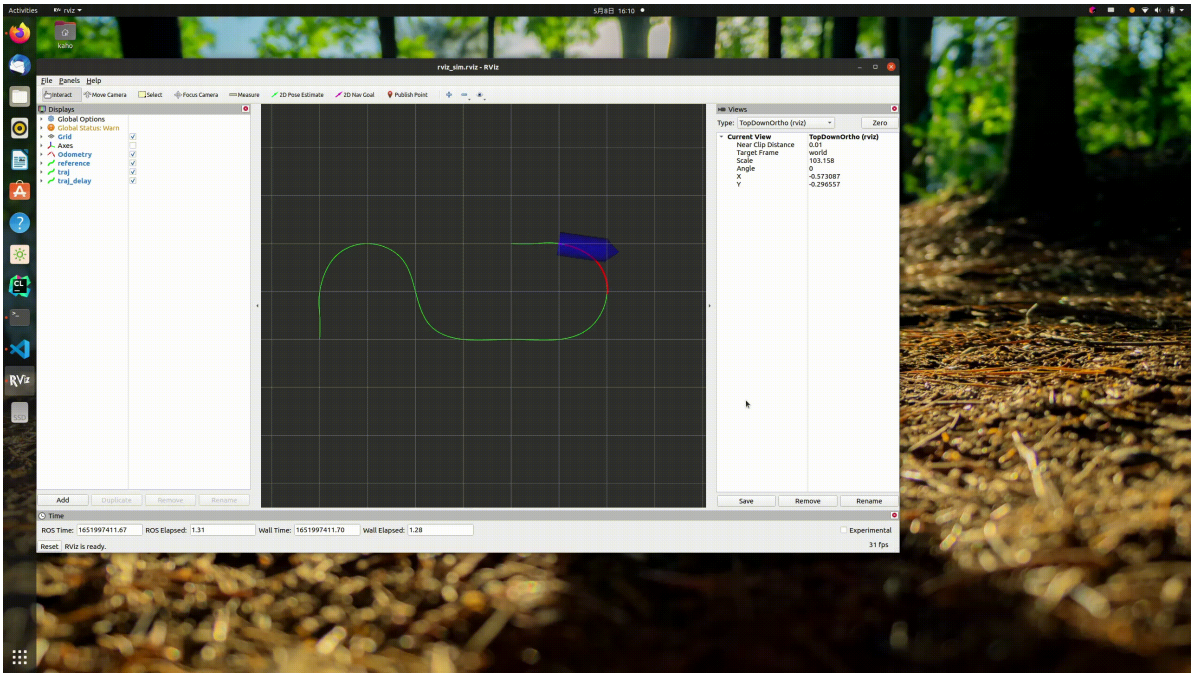
where $x \in \mathbf{R}^n$ is the optimization variable. The objective function is defined by a positive semidefinite matrix $P \in \mathbf{S}_+^n$ and vector $q \in \mathbf{R}^n$. The linear constraints are defined by matrix $A \in \mathbf{R}^{m \times n}$ and vectors l and u so that $l_i \in \mathbf{R} \cup \{-\infty\}$ and $u_i \in \mathbf{R} \cup \{+\infty\}$ for all $i \in \{1, \ldots, m\}$.

Subtitutiing the predciton model into the cost function, we have

$$U^T(BB^TQBB)U + 2(x_0AA^TQBB + GG^TQBB - q_xQBB)U$$

$q_x = [\bar{x}_1,\ \ldots, \bar{x}_N]$

2 、Result


Activities
rviz
5月8日 16:10
kaho
rviz_sim.rviz - RViz
File Panels Help
Interact
Move Camera
Select
Focus Camera
Measure
2D Pose Estimate
2D Nav Goal
Publish Point
Displays
Global Options
Global Status: Warn
Grid
Axes
Odometry
reference
traj
traj_delay
Add
Duplicate
Remove
Rename
Views
Type: TopDownOrtho (rviz)
Zero
Current View
TopDownOrtho (rviz)
Near Clip Distance 0.01
Target Frame world
Scale 103.158
Angle 0
X -0.573087
Y -0.296557
Save
Remove
Rename
Time
ROS Time: 1651997411.67
ROS Elapsed: 1.31
Wall Time: 1651997411.70
Wall Elapsed: 1.28
Experimental
Reset
RViz is ready.
31 fps