Homework 4: Kinodynamic Path Finding

Huan Chen

1. Boundary Value Problem

Instead of use fixed final state $s(T) = (p_f, v_f, a_f)$, in this homework we use $p(T) = p_f$, velocity and acceleration are free.
Modelling see slides page 26.
We have costate is solved as:

$$\lambda(t) = \frac{1}{T}\begin{bmatrix} -2\alpha \\ \alpha t + 2\beta \\ -\alpha t^2 - 2\beta t - 2\gamma \end{bmatrix}$$

For fixed final state problem, boundary condition is defined as:

$$h(s(T)) = \begin{cases} 0, & if \quad s = s(T) \\ \infty, & \text{otherwise} \end{cases}$$

For (partially)-free final state problem:

$$s_i(T) = s(T), \quad i \in I \quad \text{for fixed state}$$
$$\lambda_j(T) = \frac{\partial h(s^*(T))}{\partial s_j}, \quad j \neq i$$

So we have

$$\lambda_2(T) = 2\alpha + 2\beta\frac{1}{T} = -\Delta_v h(s^*(T)) = 0$$
$$\lambda_3(T) = -2\alpha T - 2\beta - 2\gamma\frac{1}{T} = -\Delta_a h(s^*(T)) = 0$$

We can get $\beta = -\alpha T$, $\gamma = \frac{\alpha T^2}{2}$. For position, we still have

$$\frac{1}{120}T^5\alpha + \frac{1}{24}T^4\beta + \frac{1}{6}T^3\gamma = p_f - p_0 - v_0 T - \frac{1}{2}a_0 T^2 = \Delta p$$

We can get $a = \frac{20\Delta p}{T^5}$. And obtain optimal control variable $u^*(t)$

$$\begin{aligned} u^*(t) &= -\frac{\lambda_3(t)T}{2} \\ &= \frac{1}{2}(\alpha t^2 + 2\beta t + 2\gamma) \\ &= \frac{1}{2}(\alpha t^2 - 2\alpha T t + \alpha T^2) \\ &= -\frac{10\Delta p}{T^5}t^2 + \frac{20\Delta p}{T^4}t - \frac{10\Delta p}{T^3} \end{aligned}$$

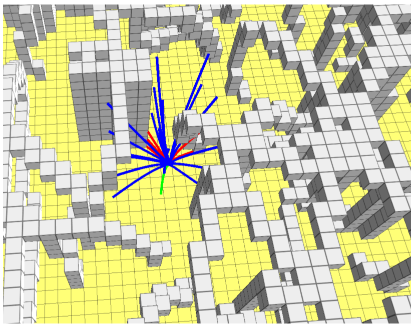

Figure 1: The result of kinodynamic path finding.

Optimal state variable $s^*(t)$:

$$s^*(t) = \begin{bmatrix} \frac{\alpha}{120}t^5 - \frac{\alpha T}{24}t^4 + \frac{\alpha T^2}{12}t^3 + \frac{a_0}{2}t^2 + v_0 t + p_0 \\ \frac{\alpha}{24}t^4 - \frac{\alpha T}{6}t^3 + \frac{\alpha T^2}{4}t^2 + a_0 t^2 + v_0 \\ \frac{\alpha}{6}t^3 - \frac{\alpha T}{2}t^2 + \frac{1}{2}\alpha T^2 t + a_0 \end{bmatrix}$$

Cost function: $J = \frac{1}{T}\int_0^T (u^*(t))^2 dt$. Solve this root finding problem can get the optimal T.

2. ROS Part

Following the given ego-graph of the linear modeled robot, the analytic expression of T:

$$J = T + \frac{a}{T} + \frac{b}{T^2} + \frac{c}{T^3}$$

We want $\frac{dJ}{dt} = 0$, this is equal to $T^4 - aT^2 - 2bT - 3c = 0$.
a, b, c are coefficients defined as following:

$$\begin{aligned} a = 4 * (&v_{x0}^2 + v_{x0}v_{xf} + v_{xf}^2 \\ &+ v_{y0}^2 + v_{y0}v_{yf} + v_{yf}^2 \\ &+ v_{z0}^2 + v_{z0}v_{zf} + v_{zf}^2) \end{aligned}$$

$$\begin{aligned} b = -12 * (&(p_{xf} - p_{x0})(v_{xf} + v_{x0}) \\ &+ (p_{yf} - p_{y0})(v_{yf} + v_{y0}) \\ &+ (p_{zf} - p_{z0})(v_{zf} + v_{z0})) \end{aligned}$$

$$c = 12 * ((p_{xf} - p_{x0})^2 + (p_{yf} - p_{y0})^2 + (p_{zf} - p_{z0})^2)$$

In ROS implement, target velocities v_{xf}, v_{yf}, v_{zf} are zero. Use the **Eigen** solver:
Eigen::PolynomialSolver<double, Eigen::Dynamic> solver;
const Eigen::PolynomialSolver<double, Eigen::Dynamic>::RootsType r = solver.roots();

3. Main Takeaways

Kinodynamic includes kinematic constraints (e.g. avoiding obstacles) and dynamic constraints (e.g. bounds on velocity, acceleration and force). Path has to follow differential constraints.

1. Why kinodynamic planning, coarse-to-fine.
2. Unicycle model, common and classic.
3. Lattice graph, sample in control vs. state space.
   - Control space: forward, no mission guidance, easy to implement, low planning efficiency
   - State space: backward, good mission guidance, hard to implement (complicated numerical optimization), high planning efficiency.
4. OBVP, the basis of state sampled lattice planning.
   - Modelling, objective function, common example is one several-order integrator.
   - Sloving, Pontryain's minimum principle.
   - Result, get optimal T and u.
5. For autonomous car, Frenet-serret frame, hybrid A*, heuristic design for various scenarios.