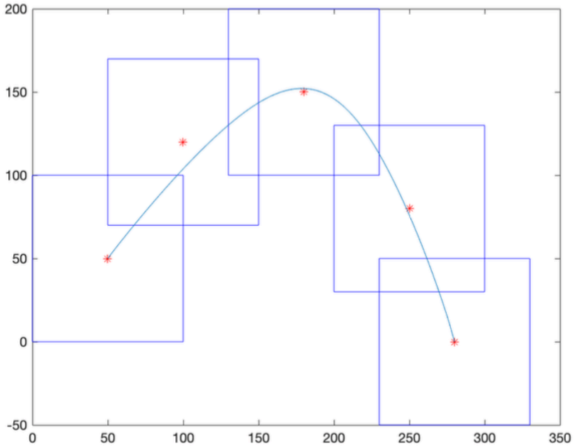
Motion Planning for Mobile Robots
第六章作业讲评

主讲人 谢晓佳

第一题

深蓝学院
shenlanxueyuan.com

- In matlab, write a corridor-constrained piecewise Bezier curve generation.
- The conversion between Bezier to monomial polynomial is given.
- The corridor is pre-defined.
- Only position needs to be constrained.
- TA provides a video tutorial.



第一题

深蓝学院 shenlanxueyuan.com

本次作业采用 **Bernstein** 多项式求解带约束的轨迹生成问题，Bernstein 多项式的基函数定义为：

$$b_i^n(t) = \binom{n}{i} \cdot t^i \cdot (1-t)^{n-i}, \quad t \in [0,1] \tag{1}$$

由基函数组成的多项式称为贝塞尔曲线，具有以下形式：

$$B_j(t) = c_j^0 b_n^0(t) + c_j^1 b_n^1(t) + \ldots + c_j^n b_n^n(t) = \sum_{i=0}^{n} c_j^i b_n^i(t) \tag{2}$$

其中系数 $[c_j^0, c_j^1, \ldots, c_j^n]$ 为曲线 j 的控制点，记为 $\mathbf{c}_j$。注意时间 t 的取值是在 $[0,1]$ 内，因此对于实际的问题需要进行归一化。特别的，对于属于 μ 中 m 段轨迹的轨迹可以写为：

$$f_\mu(t) = \begin{cases} s_1 \cdot \sum_{i=0}^{n} c_{\mu 1}^i b_n^i\left(\frac{t}{s_1}\right), & t \in [0, T_1] \\ s_2 \cdot \sum_{i=0}^{n} c_{\mu 2}^i b_n^i\left(\frac{t}{s_2}\right), & t \in [0, T_2] \\ \vdots & \vdots \\ s_m \cdot \sum_{i=0}^{n} c_{\mu m}^i b_n^i\left(\frac{t}{s_m}\right), & t \in [0, T_m] \end{cases} \tag{3}$$

其中 $s_1, \ldots, s_m$ 为归一化系数，将时间 t 转换到 $[0,1]$。注意每一段还乘上了归一化系数，这是为了获得更好的数值稳定性。 # s_i 为第 i 段的时间间隔

第一题

深蓝学院
shenlanxueyuan.com

目标函数

在本次作业中，采用和第五章一样的优化函数，即最小化 snap，则下面式子中 $k = 4$。

$$J = \sum_{\mu \in \{x,y,z\}} \int_0^T \left(\frac{d^k f_\mu(t)}{dt^k} \right)^2 dt \tag{4}$$

对于第 j 段的目标函数，化成在时间 $[0,1]$ 内标准的贝塞尔曲线 $g_{\mu j}(t)$，可以得到：

$$J_{\mu j} = \int_0^{s_j} \left(\frac{d^k f_{\mu j}(t)}{dt^k} \right)^2 dt = \int_0^1 s_j \left(\frac{s_j \cdot d^k\left(g_{\mu j}(\tau)\right)}{d(s_j \cdot \tau)^k} \right)^2 d\tau = \frac{1}{s_j^{2k-3}} \cdot \int_0^1 \left(\frac{d^k g_{\mu j}(t)}{d\tau^k} \right)^2 d\tau \quad (5)$$

s_j 为第 j 段的时间间隔，g() 为标准贝塞尔

根据上式、第五节的 $\mathbf{Q}$，以及 Bernstein polynomial 和 monomial polynomial 的关系 $p = \mathbf{M} \cdot c$，即可以得到 $\mathbf{Q}_0 = \mathbf{M}'\mathbf{Q}\mathbf{M}$。

第五章： $\mathbf{p}^T\mathbf{Q}\mathbf{p}$ $\quad$ $\mathbf{p} = \mathbf{M}\mathbf{c}$

第六章： $\mathbf{c}^T\mathbf{M}^T\mathbf{Q}\mathbf{M}\mathbf{c}$

第一题

约束条件

- Boundary Constraints:

$$\boxed{a^{l,0}_{\mu j}\cdot s_j^{(1-l)} = d^{(l)}_{\mu j}}$$

- Continuity Constraints:

$$\boxed{a^{\phi,n}_{\mu j}\cdot s_j^{(1-\phi)} = a^{\phi,0}_{\mu,j+1}\cdot s_{j+1}^{(1-\phi)},\quad a^{0,i}_{\mu j} = c^{i}_{\mu j}.}$$

- Safety Constraints:

$$\boxed{\beta^-_{\mu j} \le c^i_{\mu j}\, s \le \beta^+_{\mu j},\quad \mu\in\{x,y,z\},\quad i=0,1,2,...,n,}$$

- Dynamical Feasibility Constraints:

$$\boxed{\begin{aligned}&v_m^- \le n\cdot(c^i_{\mu j}-c^{i-1}_{\mu j}) \le v_m^+,\\ &a_m^- \le n\cdot(n-1)\cdot(c^i_{\mu j}-2c^{i-1}_{\mu j}+c^{i-2}_{\mu j})/s_j \le a_m^+\end{aligned}}$$

Stack all of these

$$\begin{aligned}\min\quad & \mathbf{c}^T\mathbf{Q}_o\mathbf{c}\\ \text{s.t.}\quad & \boxed{\mathbf{A}_{eq}\mathbf{c}=\mathbf{b}_{eq},}\\ & \boxed{\mathbf{A}_{ie}\mathbf{c}\le\mathbf{b}_{ie},}\\ & \boxed{\mathbf{c}_j\in\Omega_j,\quad j=1,2,...,m,}\end{aligned}$$

A typical convex QP formulation.

We only solve this program once to determine whether there is a qualified trajectory exists.

控制点以及 l 阶导数下的控制点的计算:

$$a^{0,i}_{\mu j} = c^i_{\mu j},\ a^{l,i}_{\mu j} = \frac{n!}{(n-l)!}\cdot\left(a^{l-1,i+1}_{\mu j} - a^{l-1,i}_{\mu j}\right),\ l\ge 1$$

https://pomax.github.io/bezierinfo/
https://en.wikipedia.org/wiki/Bernstein_polynomial

$$v_m^- \le n\cdot(c^i_{\mu j}-c^{i-1}_{\mu j}) \le v_m^+$$

$$a_m^- \le n\cdot(n-1)\cdot\frac{c^i_{\mu j}-2c^{i-1}_{\mu j}+c^{i-2}_{\mu j}}{s_j} \le a_m^+$$

$$j_m^- \le n\cdot(n-1)\cdot(n-2)\cdot\frac{c^i_{\mu j}-3c^{i-1}_{\mu j}+3c^{i-2}_{\mu j}-c^{i-3}_{\mu j}}{s_j^2} \le j_m^+$$

系数组成杨辉三角

1	n=1
1 1	n=2
1 2 1	n=3
1 3 3 1	n=4
1 4 6 4 1	n=5
1 5 10 10 5 1	n=6
1 6 15 20 15 6 1	n=7

第一题

目标函数

$$J_{\mu j}=\int_0^{s_j}\left(\frac{d^k f_{\mu j}(t)}{dt^k}\right)^2 dt=\int_0^1 s_j\left(\frac{s_j\cdot d^k\left(g_{\mu j}(\tau)\right)}{d(s_j\cdot\tau)^k}\right)^2 d\tau=\frac{1}{s_j^{2k-3}}\cdot\int_0^1\left(\frac{d^k g_{\mu j}(t)}{d\tau^k}\right)^2 d\tau \quad (5)$$

s_j 为第 j 段的时间间隔，g() 为标准贝塞尔

```
Q = [];
M = [];
M_k = getM(n_order);
for k = 1:n_seg
    %#######################################################
    % STEP 2.1 calculate Q_k of the k-th segment, minimize snap
    Q_k = [];
    t_k = 1;
    s_k = ts(k);
    for i = 4:n_order
        for l = 4:n_order
            den = i + l - 7;
            Q_k(i+1,l+1) = i*(i-1)*(i-2)*(i-3)*l*(l-1)*(l-2)*(l-3)/den*(t_k^den)/s_k^(2*4-3);
        end
    end

    Q = blkdiag(Q, Q_k);
    M = blkdiag(M, M_k);
end
```

$$J=\int_0^T\left(\frac{\mathrm{d}^k\left(s\cdot\sum c_i b^i\left(\frac{t}{s}\right)\right)}{dt^4}\right)^2 dt, t\in[0,T]$$

$$=\int_0^1 s^3\left(\frac{\mathrm{d}^k\left(\sum c_i b^i(\tau)\right)}{s^k d\tau^4}\right)^2 d\tau,\tau\in[0,1]$$

$$=s^{-(2k-3)}\int_0^1\left(\frac{\mathrm{d}^k\left(\sum p_i\tau^i\right)}{d\tau^4}\right)^2 d\tau,\tau\in[0,1]$$

$$=s^{-(2k-3)}\int_0^1 (f^4(\tau))^2 d\tau,\tau\in[0,1]$$

$$=\begin{bmatrix}\cdots\\ p_i\\ \cdots\end{bmatrix}^T\left[\cdots\frac{i(i-1)(i-2)(i-3)l(l-1)(l-2)(l-3)s^{-(2k-3)}}{i+l-7}\cdots\right]\begin{bmatrix}\cdots\\ p_i\\ \cdots\end{bmatrix}$$

第一题

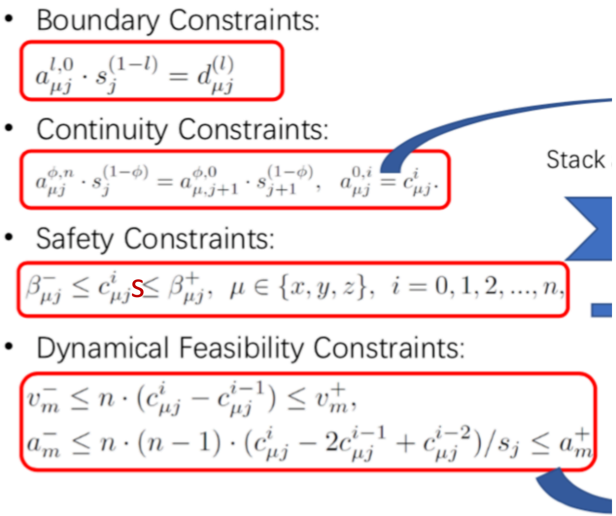
约束条件

控制点以及 l 阶导数下的控制点的计算:

$$a_{\mu j}^{0,i} = c_{\mu j}^{i}, a_{\mu j}^{l,i} = \frac{n!}{(n-l)!} \cdot \left(a_{\mu j}^{l-1,i+1} - a_{\mu j}^{l-1,i}\right), l \geq 1$$

```
n_all_poly = n_seg*(n_order+1);
%#####################################################
% STEP 2.1 p,v,a constraint in start
Aeq_start = zeros(3, n_all_poly);
beq_start = start_cond';
S = ts(1);
k = 0; Aeq_start(k+1, 1:3) = [1,0,0]*S^(1-k);
k = 1; Aeq_start(k+1, 1:3) = [-1,1,0]*n_order*S^(1-k);
k = 2; Aeq_start(k+1, 1:3) = [1,-2,1]*n_order*(n_order-1)*S^(1-k);

%#####################################################
% STEP 2.2 p,v,a constraint in end
Aeq_end = zeros(3, n_all_poly);
beq_end = end_cond';
S = ts(end);
idx = (n_seg-1)*(n_order+1) + n_order+1 - 3;
k = 0; Aeq_end(k+1, idx+(1:3)) = [0,0,1]*S^(1-k);
k = 1; Aeq_end(k+1, idx+(1:3)) = [0,-1,1]*n_order*S^(1-k);
k = 2; Aeq_end(k+1, idx+(1:3)) = [1,-2,1]*n_order*(n_order-1)*S^(1-k);
```



第一题

约束条件

控制点以及 l 阶导数下的控制点的计算:

$$a_{\mu j}^{0,i} = c_{\mu j}^{i}, a_{\mu j}^{l,i} = \frac{n!}{(n-l)!} \cdot \left(a_{\mu j}^{l-1,i+1} - a_{\mu j}^{l-1,i} \right), l \geq 1$$

- Boundary Constraints:

$$\boxed{a_{\mu j}^{l,0} \cdot s_j^{(1-l)} = d_{\mu j}^{(l)}}$$

- Continuity Constraints:

$$\boxed{a_{\mu j}^{\phi,n} \cdot s_j^{(1-\phi)} = a_{\mu,j+1}^{\phi,0} \cdot s_{j+1}^{(1-\phi)}, \quad a_{\mu j}^{0,i} = c_{\mu j}^{i}.}$$

Stack

- Safety Constraints:

$$\boxed{\beta_{\mu j}^{-} \leq c_{\mu j}^{i} \text{S} \leq \beta_{\mu j}^{+}, \quad \mu \in \{x, y, z\}, \quad i = 0, 1, 2, ..., n,}$$

- Dynamical Feasibility Constraints:

$$\boxed{\begin{aligned} &v_m^{-} \leq n \cdot (c_{\mu j}^{i} - c_{\mu j}^{i-1}) \leq v_m^{+}, \\ &a_m^{-} \leq n \cdot (n-1) \cdot (c_{\mu j}^{i} - 2c_{\mu j}^{i-1} + c_{\mu j}^{i-2})/s_j \leq a_m^{+} \end{aligned}}$$

```
%#####################################################
% STEP 2.3 position continuity constrain between 2 segments
Aeq_con_p = zeros(n_seg-1, n_all_poly);
beq_con_p = zeros(n_seg-1, 1);
k = 0;
for n = 0 : n_seg-1-1
    S = ts(n+1);
    idx = (n)*(n_order+1) + n_order+1 - 3;
    Aeq_con_p(n+1, idx+(1:3)) = [0,0,1]*S^(1-k);
    S = ts(n+1+1);
    idx = (n+1)*(n_order+1);
    Aeq_con_p(n+1, idx+(1:3)) = -[1,0,0]*S^(1-k);
end
```

```
%#####################################################
% STEP 2.4 velocity continuity constrain between 2 segments
Aeq_con_v = zeros(n_seg-1, n_all_poly);
beq_con_v = zeros(n_seg-1, 1);
k = 1;
for n = 0 : n_seg-1-1
    S = ts(n+1);
    idx = (n)*(n_order+1) + n_order+1 - 3;
    Aeq_con_v(n+1, idx+(1:3)) = [0,-1,1]*n_order*S^(1-k);
    S = ts(n+1+1);
    idx = (n+1)*(n_order+1);
    Aeq_con_v(n+1, idx+(1:3)) = -[-1,1,0]*n_order*S^(1-k);
end
```

```
%#####################################################
% STEP 2.5 acceleration continuity constrain between 2 segments
Aeq_con_a = zeros(n_seg-1, n_all_poly);
beq_con_a = zeros(n_seg-1, 1);
k = 2;
for n = 0 : n_seg-1-1
    S = ts(n+1);
    idx = (n)*(n_order+1) + n_order+1 - 3;
    Aeq_con_a(n+1, idx+(1:3)) = [1,-2,1]*n_order*(n_order-1)*S^(1-k);
    S = ts(n+1+1);
    idx = (n+1)*(n_order+1);
    Aeq_con_a(n+1, idx+(1:3)) = -[1,-2,1]*n_order*(n_order-1)*S^(1-k);
end
```

第一题

约束条件

控制点以及 l 阶导数下的控制点的计算:

$$a_{\mu j}^{0,i} = c_{\mu j}^{i}, a_{\mu j}^{l,i} = \frac{n!}{(n-l)!} \cdot \left(a_{\mu j}^{l-1,i+1} - a_{\mu j}^{l-1,i}\right), l \geq 1$$

```
n_all_poly = n_seg*(n_order+1);
%#######################################################
% STEP 3.2.1 p constraint
Aieq_p = [-eye(n_all_poly, n_all_poly); ...
           eye(n_all_poly, n_all_poly)];
bieq_p = zeros(2*n_all_poly, 1);
for k = 1 : n_seg
    t_k = ts(k);
    idx = (k-1)*(n_order+1);
    bieq_p(idx+(1:n_order+1),1) = -corridor_range(1,k)/t_k;
end
for k = 1 : n_seg
    t_k = ts(k);
    idx = n_all_poly + (k-1)*(n_order+1);
    bieq_p(idx+(1:n_order+1),1) = corridor_range(2,k)/t_k;
end
```

- Boundary Constraints:

$$a_{\mu j}^{l,0} \cdot s_j^{(1-l)} = d_{\mu j}^{(l)}$$

- Continuity Constraints:

$$a_{\mu j}^{\phi,n} \cdot s_j^{(1-\phi)} = a_{\mu,j+1}^{\phi,0} \cdot s_{j+1}^{(1-\phi)}, \quad a_{\mu j}^{0,i} = c_{\mu j}^{i}.$$

Stack

- Safety Constraints:

$$\beta_{\mu j}^{-} \leq c_{\mu j}^{i} s \leq \beta_{\mu j}^{+}, \quad \mu \in \{x, y, z\}, \quad i = 0, 1, 2, ..., n,$$

- Dynamical Feasibility Constraints:

$$v_m^- \leq n \cdot (c_{\mu j}^{i} - c_{\mu j}^{i-1}) \leq v_m^+,$$
$$a_m^- \leq n \cdot (n-1) \cdot (c_{\mu j}^{i} - 2c_{\mu j}^{i-1} + c_{\mu j}^{i-2})/s_j \leq a_m^+$$

第一题

约束条件

```
%#######################################################
% STEP 3.2.2 v constraint
Aieq_v = zeros(2*n_seg*n_order, n_all_poly);
bieq_v = ones(2*n_seg*n_order, 1) * v_max;
for k = 1 : n_seg
    r_idx = (k-1)*(n_order);
    c_idx = (k-1)*(n_order+1);
    for n = 1 : n_order
        Aieq_v(r_idx+n,c_idx+(n:n+1)) = [-1, 1]*(-1)*n_order;
    end
end
for k = 1 : n_seg
    r_idx = (k-1)*(n_order) + n_order*n_seg;
    c_idx = (k-1)*(n_order+1);
    for n = 1 : n_order
        Aieq_v(r_idx+n,c_idx+(n:n+1)) = [-1, 1]*(+1)*n_order;
    end
end

%#######################################################
% STEP 3.2.3 a constraint
Aieq_a = zeros(2*n_seg*(n_order-1), n_all_poly);
bieq_a = ones(2*n_seg*(n_order-1), 1) * a_max;
for k = 1 : n_seg
    t_k = ts(k);
    r_idx = (k-1)*(n_order-1);
    c_idx = (k-1)*(n_order+1);
    for n = 1 : n_order-1
        Aieq_a(r_idx+n,c_idx+(n:n+2)) = [1,-2,1]*(-1)*n_order*(n_order-1)/t_k;
    end
end
for k = 1 : n_seg
    t_k = ts(k);
    r_idx = (k-1)*(n_order-1) + (n_order-1)*n_seg;
    c_idx = (k-1)*(n_order+1);
    for n = 1 : n_order-1
        Aieq_a(r_idx+n,c_idx+(n:n+2)) = [1,-2,1]*(+1)*n_order*(n_order-1)/t_k;
    end
end
```

- Boundary Constraints:

$$\boxed{a^{l,0}_{\mu j} \cdot s_j^{(1-l)} = d^{(l)}_{\mu j}}$$

- Continuity Constraints:

$$\boxed{a^{\phi,n}_{\mu j} \cdot s_j^{(1-\phi)} = a^{\phi,0}_{\mu,j+1} \cdot s_{j+1}^{(1-\phi)}, \quad a^{0,i}_{\mu j} = c^i_{\mu j}.}$$

Stack

- Safety Constraints:

$$\boxed{\beta^-_{\mu j} \le c^i_{\mu j} s \le \beta^+_{\mu j}, \quad \mu \in \{x, y, z\}, \quad i = 0, 1, 2, ..., n,}$$

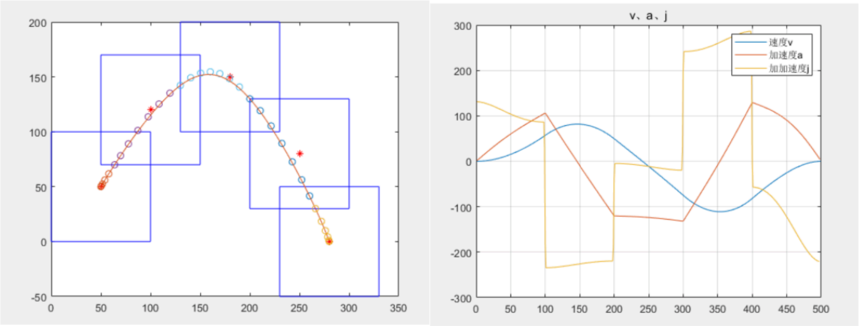
- Dynamical Feasibility Constraints:

$$\boxed{\begin{aligned} &v^-_m \le n \cdot (c^i_{\mu j} - c^{i-1}_{\mu j}) \le v^+_m, \\ &a^-_m \le n \cdot (n-1) \cdot (c^i_{\mu j} - 2c^{i-1}_{\mu j} + c^{i-2}_{\mu j})/s_j \le a^+_m \end{aligned}}$$

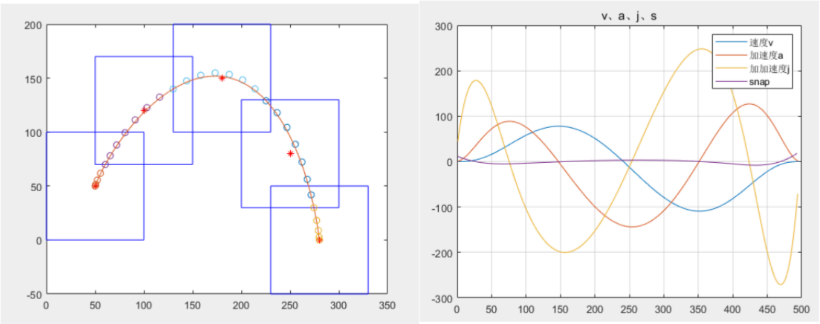
第一题

深蓝学院
shenlanxueyuan.com

1）7 阶多项式，对 jerk 没有进行约束，效果如下：

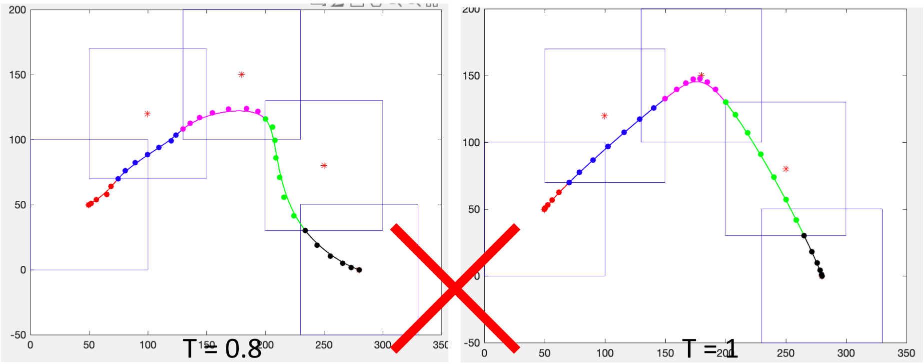


2）7 阶多项式，对 jerk 进行约束，效果如下：



对 jerk 没有约束的时候，在每条线段的交点会突变，虽然加速度在交点处有相等的约束，由于 jerk 的不连续会导致加速度在交接点不可导，但是仍然最终还是可以求出一条光滑路径。

在对 jerk 增加约束后，速度、加速度、加加速度的曲线都变得连续光滑，最终的路径也全部落在矩形框内。



深蓝学院
shenlanxueyuan.com

感谢各位聆听！

Thanks for Listening