

Minimum Snap/Jerk w/ Bezier Curve

Why use Bézier Curve to parameterize trajectory? \hookrightarrow parameterized w/ Bernstein Basis, controlled by control points

1. endpoint interpolation property
will not pass thru middle points
2. Convex Hull properties
if all control points stay within the corridor \rightarrow satisfy enclosure
3. Hodograph derivatives

↳ These could ensure the trajectory stays within constraints

Why minimum snap, should refer to differential flatness, a concept that is used in non-linear control.

Control.
Differential Flatness (see later section for quadrator flatness mapping)

for dynamical system

$$\dot{x} = f(x) + g(x)u$$

we say that the system is differential flat if there exists

$\delta \in R^m$ which can be determined by $X \in R^n$ $U \in R^m$

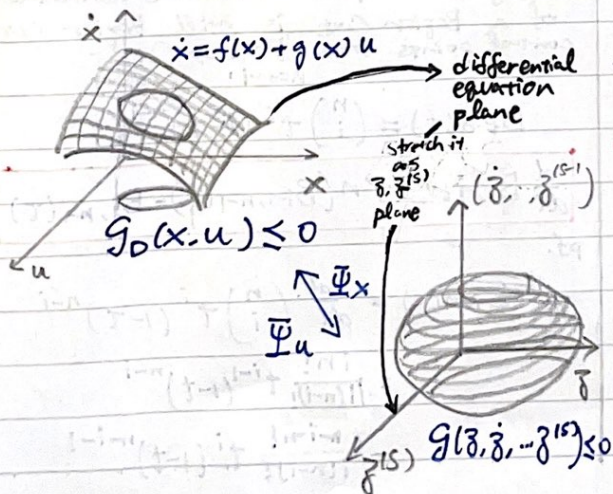
differential
outputs

$\mathcal{Z}ER^m$, could determine all state $x \in R^m$ through input $u \in R^m$.

$$x = \underline{\Psi}_x(\bar{z}, \dot{\bar{z}}, \dots, \bar{z}^{(s-1)}),$$

$$u = \Psi u(\bar{z}, \bar{z}, \dots, \bar{z}^s)$$

$\bar{\Psi}_x, \Psi_u$ are transformation processes.



for quadrotor

$$z = (r, \psi) \in \mathbb{R}^3 \times \text{SO}(2)$$

$$\downarrow \Psi \times \Psi_u$$

$$X = \{r, v, R, \omega\} \in \mathbb{R}^3 \times \mathbb{R}^3 \times \mathbb{SO}(3) \times \mathbb{R}^3$$

$$u = \{f, \tau\} \in \mathbb{R}_{\geq 0} \times \mathbb{R}^3$$

detailed in later section

\therefore So we can then plan a trajectory without differential constraints

Minimum Snap/Jerk

represent the trajectory as piecewise polynomial

$$f_j(t) = \sum_{i=0}^n P_i t^i \quad j=0 \dots m$$

- by harnessing flatness, we optimize the trajectory

- optimize it w/ snap⁽⁴⁾ || jerk⁽³⁾

$$f^{(4)}(t) = \sum_{i=4}^{\infty} i(i-1)(i-2)(i-3) t^{i-4} p_i$$

$$J(\tau) = \int_{T_{j-1}}^{T_j} (f^{(4)}(t))^2 dt$$

$$= P_1 P_2 \sum_{i=2}^n \sum_{l=2}^n \left[\frac{i(i-1)(i-2) \dots (i-l+1) l(l-1)(l-2) \dots (l-1)}{i+l-1} \right]$$

$$= P_i^T Q P_e$$

where $Q = \dots \frac{i(i-1)(i-2)(i-3)\dots(l-1)(l-2)(l-3)\dots i+l-7}{i+l-7} \dots$

$$\therefore \min J(t) \quad \text{s.t.} \quad \begin{array}{l} Ap = b \quad (\text{continuous constraints}) \\ Ap \leq b \quad (\text{limitation constraints}) \end{array}$$

Minimum Snap/Jerk w/ Constraints.

Bernstein polynomials

from monomial basis \rightarrow Bernstein basis

$$b_{i:n}(\tau) = \binom{n}{i} \tau^i (1-\tau)^{n-i}$$

$$P_j(t) = C_j^0 b_n^0(t) + C_j^1 b_n^1(t) + \dots$$

$$\dots + C_j^n b_j^n(t)$$

$$= \sum_{j=0}^n c_j \underbrace{b_n^j(t)}_{\substack{\text{bernstein} \\ \text{basis}}} \quad \swarrow \text{control points}$$

for $n+1$ points, polynomial has n order

Bernstein Polynomials (control)

From above, any polynomials:

monomial \rightarrow Bernstein basis

$$P = MC$$

recall min. snap $\min J = \min P^T Q P$

here

$$= \min C^T M^T Q M C$$

$$P_j(t) = C_j^0 b_n^0(t) + C_j^1 b_n^1(t) + \dots + C_j^n b_n^n(t) \\ = \sum_{i=0}^n C_j^i b_n^i(t) \quad b_n^i(t) = \binom{n}{i} t^i (1-t)^{n-i}$$

let $f_u(t)$ be the whole trajectory

$$f_u(t) = \begin{cases} s_1 \sum_{i=0}^n C_{u1}^i b_n^i\left(\frac{t-T_0}{s_1}\right), & t \in [T_0, T_1] \\ s_2 \sum_{i=0}^n C_{u2}^i b_n^i\left(\frac{t-T_1}{s_2}\right), & t \in [T_1, T_2] \\ \vdots \\ s_m \sum_{i=0}^n C_{um}^i b_n^i\left(\frac{t-T_{m-1}}{s_m}\right), & t \in [T_{m-1}, T_m] \end{cases}$$

with piece trajectory

$$\tau = \frac{t - T_j}{s_{j+1}} \quad \text{normalization}$$

$$\tau \in [0, 1]$$

$$J = \sum_{u \in \mathcal{U}} \int_0^T \left(\frac{d^k f_u(t)}{dt^k} \right)^2 dt$$

on u axis, jth trajectory

$$s dt = d\tau$$

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

e.g. for $t \in [0, T]$

$$J = \int_0^T \left(\frac{d^k (s \cdot \sum C_i b^i(\frac{t}{s}))}{dt^k} \right)^2 dt \\ = \int_0^1 \left(\frac{d^k (s \cdot \sum C_i b^i(\tau))}{s^k d\tau^k} \right)^2 s d\tau \\ = \int_0^1 s^{3-2k} \left(\frac{d^k \sum C_i b^i(\tau)}{d\tau^k} \right)^2 d\tau \\ = \int_0^1 s^{3-2k} \left(\frac{d^k \sum P_i \tau^i}{d\tau^k} \right)^2 d\tau$$

get Q from $P^T Q P$ or $C^T M^T Q M C$

$$Q = \begin{bmatrix} \dots & \frac{1!(1-1)(1-2)(1-3)\dots(1-k+1)(k-2)(k-3)\dots(1-2k+3)}{i!k-7} s^{-2k+3} & \dots \end{bmatrix}$$

get M from $C^T M^T Q M C$

pascal triangle

now we have Q & M

minimize objectives: $\min J = C^T M^T Q M C$

minimize conditions: $Ac = b$
 $Ac \leq b$

$$C \in \mathbb{R}^{(n_{order}+1) \times m}$$

Bézier Curve Homograph

Homograph implies that the derivatives of a Bézier Curve is still Bézier curve
control points from $P_i \rightarrow Q_i$
 $i=0 \sim n+1 \quad i=0 \sim n$

$$B_{i,n}(t) = \binom{n}{i} t^i (1-t)^{n-i}$$

$$\frac{d}{dt} B_{i,n}(t) = n(B_{i-1,n-1}(t) - B_{i,n-1}(t))$$

pf.

$$\frac{d}{dt} B_{i,n}(t) = \frac{d}{dt} \binom{n}{i} t^i (1-t)^{n-i} \\ = \frac{i n!}{i!(n-i)!} t^{i-1} (1-t)^{n-i} \\ - \frac{(n-i)n!}{i!(n-i)!} t^i (1-t)^{n-i-1}$$

$$\begin{aligned}
 B_{i,n}(t) &= \frac{n!}{i!(n-i)!} t^{i-1} (1-t)^{n-i} - \frac{(n-1)!}{i!(n-i)!} t^i (1-t)^{n-i-1} \\
 &= \frac{n(n-1)!}{i!(n-i)!} t^{i-1} (1-t)^{n-i} - \frac{n(n-1)!}{i!(n-i)!} t^i (1-t)^{n-i-1} \\
 &= n \left[\frac{(n-1)!}{i!(n-i)!} t^{i-1} (1-t)^{n-i} - \frac{(n-1)!}{i!(n-i)!} t^i (1-t)^{n-i-1} \right] \\
 &= n(B_{i-1,n-1}(t) - B_{i,n-1}(t))
 \end{aligned}$$

$$F(t) = \sum_{i=0}^n B_{i,n}(t) P_i$$

$$\frac{d}{dt} F(t) = \sum_{i=0}^{n-1} B_{i,n-1}(t) Q_i \quad Q_i = n(P_{i+1} - P_i)$$

PS.

let $n=2$

$$F(t) = \sum_{i=0}^2 B_{i,2}(t) P_i$$

$$\begin{aligned}
 &= B_{0,2}(t) P_0 \\
 &+ B_{1,2}(t) P_1 \\
 &+ B_{2,2}(t) P_2
 \end{aligned}$$

$$\begin{aligned}
 \frac{d}{dt} F(t) &= 2(B_{0,1}(t) - B_{0,2}(t)) P_0 \\
 &+ 2(B_{0,1}(t) - B_{1,1}(t)) P_1 \\
 &+ 2(B_{1,1}(t) - B_{2,1}(t)) P_2 = 0
 \end{aligned}$$

when $n=1$, derivative of $F(t)$, $n=2$

$$\begin{aligned}
 F(t) &= \sum_{i=0}^1 B_{i,1}(t) Q_i \\
 &= B_{0,1}(t) Q_0 \\
 &+ B_{1,1}(t) Q_1 = 0
 \end{aligned}$$

from ①

$$\frac{d}{dt} F(t) = (-2P_0 + 2P_1) B_{0,1}(t) + (-2P_1 + 2P_2) B_{1,1}(t)$$

$$= 2(P_1 - P_0) B_{0,1}(t) + 2(P_2 - P_1) B_{1,1}(t)$$

$$\text{from ②} \quad = Q_0 B_{0,1}(t) + Q_1 B_{1,1}(t)$$

$$Q_0 = 2(P_1 - P_0)$$

$$Q_1 = 2(P_2 - P_1)$$

$$Q_i = n(P_{i+1} - P_i)$$

e.g. $P_0 \ P_1 \ P_2 \ P_3 \ P_4 \quad n=4 \quad \text{pts} \times 5$

$$\frac{d}{dt} \left(n(P_1 - P_0), n(P_2 - P_1), n(P_3 - P_2), n(P_4 - P_3) \right) \quad \text{pts} \times 4$$

$$\frac{d}{dt} \left((n-1) \begin{pmatrix} n(P_2 - P_1) - n(P_1 - P_0) \\ n(P_3 - P_2) - n(P_2 - P_1) \\ n(P_4 - P_3) - n(P_3 - P_2) \end{pmatrix} \right) \quad \text{pts} \times 3$$

Equality Condition

starting condition

ending condition

continuous condition

$$Ac = b$$

$$C \in R_{(n\text{-order}+1) \times m}$$

n -order:
highest order of
polynomial
 m :
no. of parameters

e.g. $n\text{-order} = 7$
 $m = 5$

$$C \in R_{(7+1) \times 5} = R_{40}$$

$$A \in R_{dx(n\text{-order}+1) \times m} = R_{dx40}$$

$$b \in R_d \rightarrow dx \text{ constraints}$$

starting condition, ending condition $j=0$
 $j=f$

$$a_{0,j} = c_0 \quad (u_j) \quad a_{f,j} = c_1 \quad a_{f,j} = c_2$$

$$P = a_{f,j} (S_j)^f$$

$$V = \frac{d}{dt} a_{f,j} (S_j)^f = n(a_{f,j} - a_{0,j})$$

$$a = \frac{d^2}{dt^2} a_{f,j} (S_j)^f = n(n-1)(a_{f,j} - a_{0,j}) \cdot (a_{f,j} - a_{0,j})$$

Write everything into matrix from above

e.g. $n\text{-order} = 7$
 $m = 5$
 $S_j = t$

$$Ac = b$$

$$\begin{bmatrix} 1t & 0 & 0 \\ -7t^0 & 7t^0 & 0 \\ 42t^1 & -84t^1 & 42t^1 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} c_0 \\ c_1 \\ c_2 \\ \vdots \\ c_8 \end{bmatrix} = \begin{bmatrix} P \\ V \\ a \\ \vdots \\ 0 \end{bmatrix}$$

continuous condition

$$P_{n\text{-order}+1,j} = P_{0,j+1} \quad j = 0, 1, 2, \dots, m$$

$$V_{n\text{-order}+1,j} = V_{0,j+1} \quad j = 0, 1, 2, \dots, m$$

$$a_{n\text{-order}+1,j} = a_{0,j+1} \quad j = 0, 1, 2, \dots, m$$

$$\therefore \begin{cases} P_{n\text{-order}+1,j} - P_{0,j+1} = 0 \\ V_{n\text{-order}+1,j} - V_{0,j+1} = 0 \\ a_{n\text{-order}+1,j} - a_{0,j+1} = 0 \end{cases}$$

$$j = 0, 1, 2, \dots, m$$

$$= 0$$

$$S_j = t$$

Equality condition (cont'd)

$$\begin{cases} P_{n\text{-order}+1,j} - P_{0,j+1} = 0 \\ V_{n\text{-order}+1,j} - V_{0,j+1} = 0 \\ A_{n\text{-order}+1,j} - A_{0,j+1} = 0 \end{cases} \quad \left\{ \begin{array}{l} \frac{d}{dt} \\ \frac{d^2}{dt^2} \end{array} \right\} \quad \text{Bezier Curve}$$

$$\Rightarrow \begin{cases} a_{n\text{-order}+1,j} - a_{0,j+1} = 0 \\ \frac{d}{dt} a_{n\text{-order}+1,j} - \frac{d}{dt} a_{0,j+1} = 0 \\ \frac{d^2}{dt^2} a_{n\text{-order}+1,j} - \frac{d^2}{dt^2} a_{0,j+1} = 0 \end{cases} \quad \left\{ \begin{array}{l} \text{first pt of } j+1 \\ \parallel \\ \text{last pt of } j \end{array} \right\}$$

$$\Rightarrow \begin{cases} a_{n\text{-order}+1,j} - a_{0,j+1} = 0 \\ n(a_{n\text{-order}+1,j} - a_{0,j+1}) - n(a_{0,j+1} - a_{0,j+1}) = 0 \\ n(n-1)[(a_{n\text{-order}+1,j} - a_{0,j+1}) - (a_{0,j+1} - a_{0,j+1})] \\ -n(n-1)[(a_{0,j+1} - a_{0,j+1}) - (a_{0,j+1} - a_{0,j+1})] = 0 \end{cases}$$

↳ write everything into matrix

e.g. $n\text{-order} = 7$
 $m = 5$
 condition @ $j = 1, 2$

4 connection conditions
 ↓
 12 conditions

$$\begin{bmatrix} \dots & 1 & -1 & \dots \\ \dots & -7 & 7 & -7 & \dots \\ 42 & -84 & 42 & -42 & 84 & -42 \\ \dots & \dots & \dots & \dots & \dots & \dots \end{bmatrix} \begin{bmatrix} C_0 \\ \vdots \\ C_8 \\ \vdots \\ C_{40} \end{bmatrix} = \begin{bmatrix} 0 \\ \vdots \\ 0 \end{bmatrix}$$

$R_{12 \times 40}$

$R_{40 \times 1}$

R_{12}

w, w/o t^n
 does not matter
 as $AC = 0$

$$Ac = b$$

$$A \in R_{d \times 40}$$

$$c \in R_{40}$$

$$b \in R_d$$

d: no. of conditions
 starting
 ending
 continuous

Inequality condition (cont'd)

safety corridor
 dynamic constraints

$$Ac \leq b$$

- main reason for us to use bezier curve is that we can **CONFINE** all control points within our desired range

- all expression should be in the form of $Ac \leq b$

$$A \in R_{d \times (n\text{-order}+1) \times m}$$

$$C \in R_{(n\text{-order}+1) \times m}$$

$$b \in R_d$$

$d \Rightarrow$ no. of conditions

safety corridor

let $a \leq x \leq b$

$$c \leq y \leq d$$

$$e \leq z \leq f$$

$$\Rightarrow \begin{cases} x \leq b \\ -x \leq -a \\ y \leq d \\ -y \leq -c \\ z \leq f \\ -z \leq -e \end{cases}$$

write everything in matrix

$$Ac \leq b$$

express everything in terms of control points

$$\Rightarrow t^1 \begin{bmatrix} 1 & & & \\ & 1 & & \\ & & \ddots & \\ & & & 1 \end{bmatrix} \begin{bmatrix} C_0 \\ \vdots \\ C_{(n\text{-order}+1) \times m} \end{bmatrix} \leq b$$

ieq-p

$$\Rightarrow t^0 \begin{bmatrix} -7 & 7 & & \\ & 7 & 7 & \\ & 7 & -7 & \\ & & 7 & -7 \end{bmatrix} \begin{bmatrix} C_0 \\ \vdots \\ C_{(n\text{-order}+1) \times m} \end{bmatrix} \leq b$$

ieq-v

$$\Rightarrow t^1 \begin{bmatrix} 42 & -84 & 42 & & \\ & 42 & -84 & 42 & \\ & & \ddots & \ddots & \\ & & & -42 & 84 & -42 \\ & & & & -42 & 84 & -42 \end{bmatrix} \begin{bmatrix} C_0 \\ \vdots \\ C_{(n\text{-order}+1) \times m} \end{bmatrix} \leq b$$

ieq-v

e.g. $n\text{-order} = 7$
 $m = 5$

no. of p ctrl-ptcs = 80
 $= 70$

$$A \in R_{210 \times 40}$$

$$C \in R_{40}$$

$$b \in R_{210}$$