Minimum Jerk

一些假设与背景

下列问题都是在单一维度进行的！比如x方向，y方向或者z方向，如果想要综合考虑，只需要把代价函数和约束加起来即可。

对于轨迹优化（jerk）问题，一般假设轨迹为五次多项式：

$$x(t) = p_5 t^5 + p_4 t^4 + p3 t^3 + p_2 t^2 + p_1 t + p_0 \tag{1}$$

这里的次数选择依据于：

如果只有一段轨迹，即只有始末两个节点，并且对jerk进行最小化，那么状态量就是x，v，a，此时一共有6个约束参数，等价于有6个方程。那么如果想要方程可解，**需要有大于等于6个参数**，如果大于6个就是欠定方程组（一般无穷组解），小于6个就是超定方程组（一般无解）。

因此多项式最高次数为5，此时刚好有6个参数（当然大于6个也可以）。

实际中，在始末两点之间可能指定通过多个路径点，但是在这些中间点中除了位置外，其他状态并不做硬约束，而是在优化中得到他们的值。

凸优化问题

损失函数最小化jerk或者最小化snap其实都可以，这里以最小化jerk中最简单的形式举例，不考虑其他软约束。待优化函数为：

$$J_j(T) = \int_{T_j-1}^{T_j} [x^{(3)}(t)]^2 dt \tag{2}$$

其中j代表第j段路径。

$$x^{(3)}(t) = \sum_{i\geq 3} i(i-1)(i-2)t^{i-3} p_i$$

$$[x^{(3)}(t)]^2 = \sum_{i\geq 3, l\geq 3} i(i-1)(i-2)l(l-1)(l-2)t^{i+l-6} p_i p_l$$

$$J_j(T) = \int_{T_j-1}^{T_j} [x^{(3)}(t)]^2 dt$$

$$J_j(T) = \sum_{i\geq 3, l\geq 3} \frac{i(i-1)(i-2)l(l-1)(l-2)}{i+l-5} (T_j^{i+l-5} - T_{j-1}^{i+l-5}) p_i p_j$$

$$J_j(T) = \begin{bmatrix} p3 \\ p4 \\ p5 \end{bmatrix}^T * \begin{bmatrix} A_{33} & A_{34} & A_{35} \\ A_{43} & A_{44} & A_{45} \\ A_{53} & A_{54} & A_{55} \end{bmatrix} * \begin{bmatrix} p3 \\ p4 \\ p5 \end{bmatrix}$$

那么整段路径的损失函数为：

$$J(T) = \begin{bmatrix} \vec{p_1} \\ \vec{p_2} \\ . \\ . \\ . \\ \vec{p_M} \end{bmatrix}^T * \begin{bmatrix} Q_1 & 0 & \dots & 0 \\ 0 & Q_2 & \dots & 0 \\ . & . & \dots & . \\ 0 & 0 & 0 & Q_M \end{bmatrix} * \begin{bmatrix} \vec{p_1} \\ \vec{p_2} \\ . \\ . \\ . \\ \vec{p_M} \end{bmatrix}$$

约束条件有两个：节点状态和连续性

$$
\begin{aligned}
& f_j^{(k)}(T_j) = x_j^{(k)} \\
\Rightarrow & \sum_{i \geq k} \frac{i!}{(i-k)!} T_j^{i-k} p_{j,i} = x_{T,j}^{(k)} \\
\Rightarrow & \begin{bmatrix} \cdots & \frac{i!}{(i-k)!} T_j^{i-k} & \cdots \end{bmatrix} \begin{bmatrix} \vdots \\ p_{j,i} \\ \vdots \end{bmatrix} = x_{T,j}^{(k)} \\
\Rightarrow & \begin{bmatrix} \cdots & \frac{i!}{(i-k)!} T_{j-1}^{i-k} & \cdots \\ \cdots & \frac{i!}{(i-k)!} T_j^{i-k} & \cdots \end{bmatrix} \begin{bmatrix} \vdots \\ p_{j,i} \\ \vdots \end{bmatrix} = \begin{bmatrix} x_{0,j}^{(k)} \\ x_{T,j}^{(k)} \end{bmatrix} \\
\Rightarrow & \mathbf{A}_j \mathbf{p}_j = \mathbf{d}_j
\end{aligned}
$$

$$
\begin{aligned}
& f_j^{(k)}(T_j) = f_{j+1}^{(k)}(T_j) \\
\Rightarrow & \sum_{i \geq k} \frac{i!}{(i-k)!} T_j^{i-k} p_{j,i} - \sum_{l \geq k} \frac{l!}{(l-k)!} T_j^{l-k} p_{j+1,l} = 0 \\
\Rightarrow & \begin{bmatrix} \cdots & \frac{i!}{(i-k)!} T_j^{i-k} & \cdots & -\frac{l!}{(l-k)!} T_j^{l-k} & \cdots \end{bmatrix} \begin{bmatrix} \vdots \\ p_{j,i} \\ \vdots \\ p_{j+1,l} \\ \vdots \end{bmatrix} = 0 \\
\Rightarrow & \begin{bmatrix} \mathbf{A}_j & -\mathbf{A}_{j+1} \end{bmatrix} \begin{bmatrix} \mathbf{p}_j \\ \mathbf{p}_{j+1} \end{bmatrix} = 0
\end{aligned}
$$

这两个条件可以合并到一个矩阵，因此整个优化问题为：

$$
min \begin{bmatrix} \vec{p_1} \\ \vec{p_2} \\ . \\ . \\ . \\ \vec{p_M} \end{bmatrix}^T * \begin{bmatrix} Q_1 & 0 & \dots & 0 \\ 0 & Q_2 & \dots & 0 \\ . & . & \dots & . \\ 0 & 0 & 0 & Q_M \end{bmatrix} * \begin{bmatrix} \vec{p_1} \\ \vec{p_2} \\ . \\ . \\ . \\ \vec{p_M} \end{bmatrix}
$$

$$
s.t. A_{eq} * \begin{bmatrix} \vec{p_1} \\ \vec{p_2} \\ . \\ . \\ . \\ \vec{p_M} \end{bmatrix} = d_{eq}
$$

这是一个标准的凸优化中的QP问题。

由于此时的变量是不具有物理意义的，只是多项式的次数，这样是数值不稳定的，因此将它们转换为具有物理意义的各阶导数。

$$设M_j\vec{p_j} = \vec{d_j}$$

$$x(t) = p_5 t^5 + p_4 t^4 + p3t^3 + p_2 t^2 + p_1 t + p_0$$

$$x'(t) = 5p_5 t^4 + 4p_4 t^3 + 3p3t^2 + 2p_2 t + p_1$$

$$x''(t) = 20p_5 t^3 + 12p_4 t^2 + 6p_3 t + 2p_2$$

$$由于x(0) = p_0, x'(0) = v_0, x''(0) = a_0, x(T) = p_T, x'(T) = v_T, x''(T) = a_T, 因此有$$

$$M = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 2 & 0 & 0 \\ T^5 & T^4 & T^3 & T^2 & T & 1 \\ 5T^4 & 4T^3 & 3T^2 & 2T & 1 & 0 \\ 20T^3 & 12T^2 & 6T & 2 & 0 & 0 \end{bmatrix}$$

$$J = \begin{bmatrix} \vec{d_1} \\ \vec{d_2} \\ . \\ . \\ . \\ \vec{d_M} \end{bmatrix}^T * \begin{bmatrix} M_1 & 0 & \dots & 0 \\ 0 & M_2 & \dots & 0 \\ . & . & \dots & . \\ 0 & 0 & 0 & M_M \end{bmatrix}^{-T} * \begin{bmatrix} Q_1 & 0 & \dots & 0 \\ 0 & Q_2 & \dots & 0 \\ . & . & \dots & . \\ 0 & 0 & 0 & Q_M \end{bmatrix} * \begin{bmatrix} M_1 & 0 & \dots & 0 \\ 0 & M_2 & \dots & 0 \\ . & . & \dots & . \\ 0 & 0 & 0 & M_M \end{bmatrix}^{-1} * \begin{bmatrix} \vec{d_1} \\ \vec{d_2} \\ . \\ . \\ . \\ \vec{d_M} \end{bmatrix}$$

此时就可以作为一个凸优化中标准的QP问题求解了，但高飞还讲了一种closed form的解法，**就是把约束量隐含在代价函数中，进而就可以作为无约束的问题求解，相当于求解二次函数极值点了**

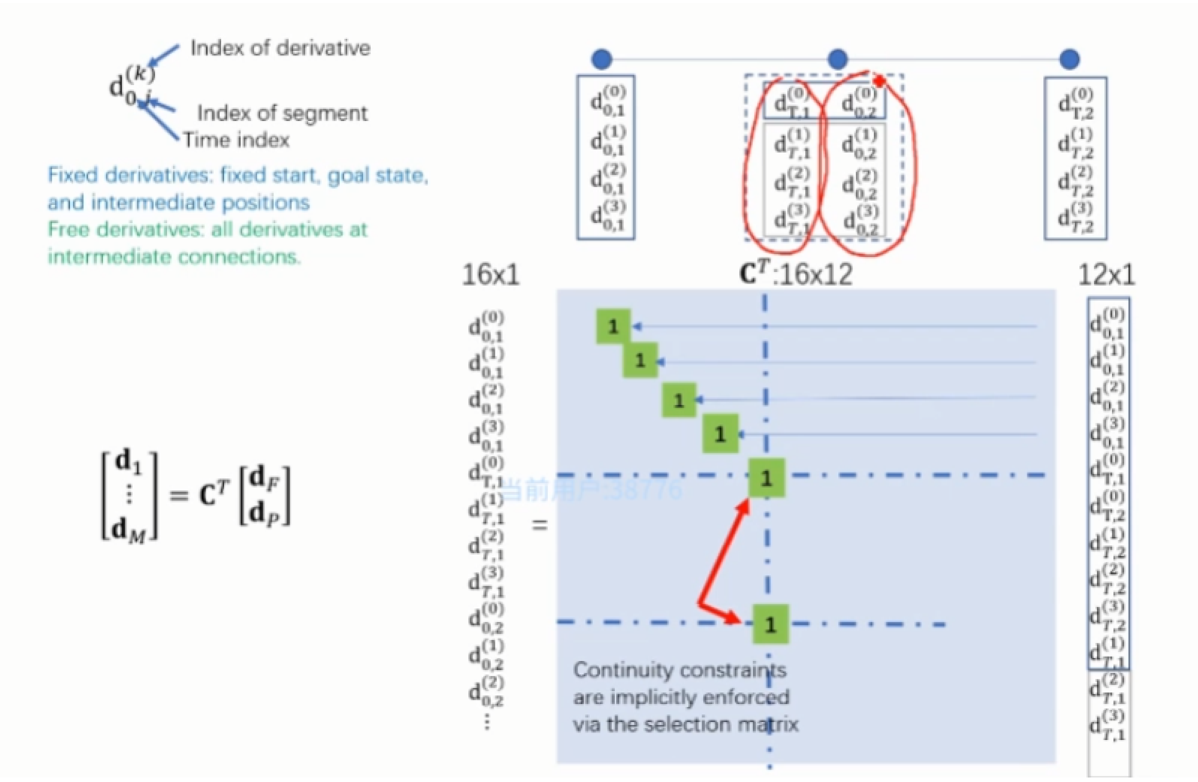
$$设\vec{d_F}是约束条件中的约束状态量，\vec{d_P}是在优化过程中确定的自由量，比如中间节点的速度和加速度$$

$$C^T \begin{bmatrix} \vec{d_F} \\ \vec{d_P} \end{bmatrix} = \begin{bmatrix} \vec{d_1} \\ \vec{d_2} \\ . \\ . \\ . \\ \vec{d_M} \end{bmatrix}$$

$$注意，由于连续性约束，因此等式右边的状态量是有重复的，比如第一段的T时刻和第二段的0时刻。$$

$$因此经过C^T转换后，两个向量的维度不一样。$$

$$C^T的求法也很简单，把两个向量摆在等式左右两边，对照着把1填在对应的位置就行。$$



由于dF中的值并不是变量，而是根据约束条件给定的，并且在删除重复状态量的过程中，也是隐含了满足连续性条件，因此在这个方程中其实已经隐含了所有的约束，此时就可以转化为一个无约束问题。

$$J = \begin{bmatrix} \vec{d}_F \\ \vec{d}_P \end{bmatrix}^T CM^{-T}QM^{-1}C^T \begin{bmatrix} \vec{d}_F \\ \vec{d}_P \end{bmatrix} = \begin{bmatrix} \vec{d}_F \\ \vec{d}_P \end{bmatrix}^T \begin{bmatrix} R_{FF} & R_{FP} \\ R_{PF} & R_{PP} \end{bmatrix} \begin{bmatrix} \vec{d}_F \\ \vec{d}_P \end{bmatrix}$$

$$J = d_F^T R_{FF} d_F + d_F^T R_{FP} d_p + d_p^T R_{PF} d_F + d_P^T R_{PP} d_P$$

$$d_P^* = -R_{PP}^{-1} R_{FP}^T d_F$$

工程上的细节处理

归一化

$$f(t) = \begin{cases} f_1(t) \doteq \sum_{i=0}^{N} p_{1,i} t^i & T_0 \le t \le T_1 \\ f_2(t) \doteq \sum_{i=0}^{N} p_{2,i} t^i & T_1 \le t \le T_2 \\ \vdots & \vdots \\ f_M(t) \doteq \sum_{i=0}^{N} p_{M,i} t^i & T_{M-1} \le t \le T_M \end{cases}$$

如果采用累积时间作为每段时间的记法，那么当时间累积很大时，运算结果会非常大，数值非常不稳定，因此采用相对时间记录。归一化之后并不会影响多项式参数的计算，只是将多项式压缩了一下，形状不变，可以在得到多项式系数后再拉伸回来。

$$
f(t)=\begin{cases}
\sum_{i=0}^{N} p_{1,i}\left(\frac{t-T_0}{T_1-T_0}\right)^i & T_0 \le t \le T_1 \\
\sum_{i=0}^{N} p_{2,i}\left(\frac{t-T_1}{T_2-T_1}\right)^i & T_1 \le t \le T_2 \\
\vdots & \vdots \\
\sum_{i=0}^{N} p_{M,i}\left(\frac{t-T_{M-1}}{T_M-T_{M-1}}\right)^i & T_{M-1} \le t \le T_M
\end{cases}
$$

同理如果空间体积很大的话，也可以将空间归一化，在一个类似沙盘上求解。

- Problem scale (spatial) normalization
  - If the problem is underlying for large-scale scene.
  - Such as waypoints with $\boldsymbol{x} = \mathbf{100.0}\ \boldsymbol{m}$
  - Consider solve a tiny problem (a sandbox), and re-scale the solution back.

维度的独立问题

对于无人机来说，由于三个维度是解耦的，因此在优化函数中不存在软约束时，可以三个维度独立求解。但如果加入软约束，需要将三个维度信息一起代入软约束求值，此时就需要将三个维度的损失函数求和。

Is closed-form solution always better?

作者在论文中说closed-form的求解是更好的，但这涉及矩阵的求逆，因此如果计算资源不够的话，还是使用标准的QP问题求解方式。

是否可以用多项式作为所有问题的路径假设?

几乎是，但有一些情况多项式不是最优的。当损失函数是两个状态量加权时，比如：

$$J=\int_0^T \rho_1 \cdot jerk^2(t)+\rho_2 \cdot snap^2(t)dt$$

此时多项式就不是一个最优的假设，但也是逼近最优的，所以理论上用多项式也可以。