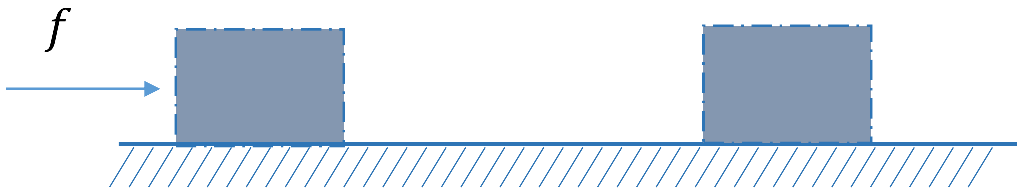
# 01 轨迹优化简介



- **从一个简单的例子说起**

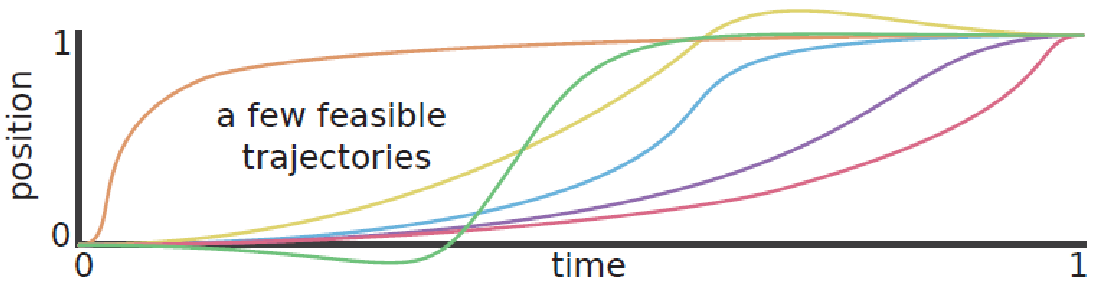
  - ■ 问题
  - ➢ 将光滑水平面上的滑块在力$f$的作用下从$A$点转移到$B$点



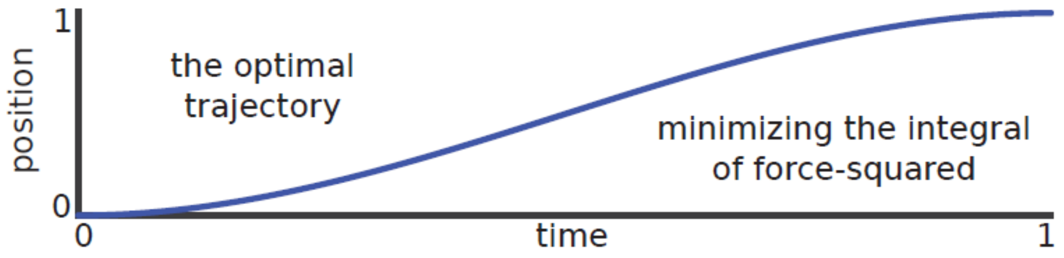

| $A$ | $B$ |
|---|---|
| $t = 0$ | $t = 1$ |
| $x = 0$ | $x = 1$ |
| $v = 0$ | $v = 0$ |

  - ■ 可行解
  - ➢ 非唯一

  - ■ 最优解
  - ➢ 唯一




可行解轨迹




最优轨迹

Kelly M. An introduction to trajectory optimization: How to do your own direct collocation[J]. SIAM Review, 2017, 59(4): 849-904.

## ● 从一个简单的例子说起

■ 约束

➢ 轨迹优化问题中所有的问题要求，也就是所谓的约束；滑块问题的约束有两个：

➢ System Dynamics

$$\dot{x} = \nu, \qquad \dot{\nu} = u, \qquad \textbf{system dynamics.}$$

➢ Boundary Conditions

$$\begin{aligned} x(0) &= 0, & x(1) &= 1, \\ \nu(0) &= 0, & \nu(1) &= 0, \end{aligned}$$

■ 可行解

➢ 满足所有约束的解，被称为可行解；产生可行轨迹的控制集称为容许控制

■ 最优解

➢ 轨迹优化是指寻找可行轨迹中的最优轨迹；通常我们形式化一个目标函数来描述最优轨迹；

➢ 滑块问题常用的目标函数：最小力平方和最小绝对功

$$\min_{u(t),\,x(t),\,\nu(t)} \int_0^1 u^2(\tau)\, d\tau, \qquad \textbf{minimum force squared,}$$

$$\min_{u(t),\,x(t),\,\nu(t)} \int_0^1 \big|u(\tau)\,\nu(\tau)\big|\, d\tau, \qquad \textbf{minimum absolute work.}$$

## ● 轨迹优化问题

■ 目标函数

- (1) Mayer形式：仅包含边界项
- (2) Lagrange形式：仅包含路径积分项
- (3) Bolza形式：包含边界项和路径积分项

$$\boxed{\min_{t_0, t_F, \boldsymbol{x}(t), \boldsymbol{u}(t)} \underbrace{J(t_0, t_F, \boldsymbol{x}(t_0), \boldsymbol{x}(t_F))}_{\text{Mayer Term}} + \underbrace{\int_{t_0}^{t_F} w(\tau, \boldsymbol{x}(\tau), \boldsymbol{u}(\tau))\, d\tau}_{\text{Lagrange Term}}.}$$

■ 约束条件

- (1) System Dynamics
- (2) path constraints
- (3) boundary constraints
- (4) path bound on states
- (5) path bound on controls
- (6) bounds on initial and final time
- (7) bound on initial state
- (8) bound on final state

$$\boxed{\dot{\boldsymbol{x}}(t) = \boldsymbol{f}(t, \boldsymbol{x}(t), \boldsymbol{u}(t))}$$

$$\boxed{\boldsymbol{h}(t, \boldsymbol{x}(t), \boldsymbol{u}(t)) \le \boldsymbol{0},}$$

$$\boxed{\boldsymbol{g}(t_0, t_F, \boldsymbol{x}(t_0), \boldsymbol{x}(t_F)) \le \boldsymbol{0}}$$

$$\boxed{\begin{aligned} &\boldsymbol{x}_{\text{low}} \le \boldsymbol{x}(t) \le \boldsymbol{x}_{\text{upp}}, \\ &\boldsymbol{u}_{\text{low}} \le \boldsymbol{u}(t) \le \boldsymbol{u}_{\text{upp}} \end{aligned}}$$

$$\boxed{\begin{aligned} &t_{\text{low}} \le t_0 < t_F \le t_{\text{upp}}, \\ &\boldsymbol{x}_{0,\text{low}} \le \boldsymbol{x}(t_0) \le \boldsymbol{x}_{0,\text{upp}}, \\ &\boldsymbol{x}_{F,\text{low}} \le \boldsymbol{x}(t_F) \le \boldsymbol{x}_{F,\text{upp}}, \end{aligned}}$$

- **变分法**

  - 目标函数（优化准则）$$J=\int_0^1 u^2(\tau)d\tau=\int_0^1 \ddot{x}^2(\tau)d\tau$$

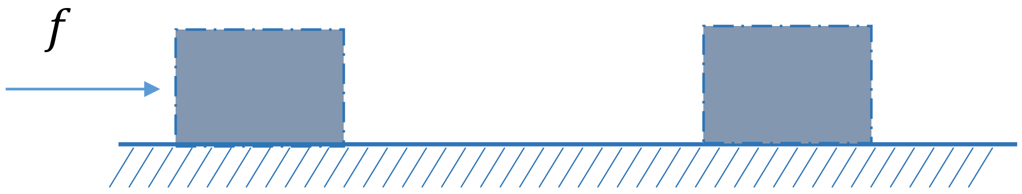
  - Euler-Lagrange方程 $$\mathcal{L}(t,x,\dot{x},\ddot{x})=\mathcal{L}(\ddot{x})=\ddot{x}^2$$

  - Lagrange函数

  - 求解结果 $$\frac{\partial\mathcal{L}}{\partial x}-\frac{d}{dt}\frac{\partial\mathcal{L}}{\partial\dot{x}}-\frac{d^2}{dt^2}\frac{\partial\mathcal{L}}{\partial\ddot{x}}=0$$

得： $x^{(4)}=0$，也就是说最优轨迹具有三次多项式的形式！

代入边界条件可以得到：$$x^*(t)=3t^2-2t^3$$




| $A$ | $B$ |
|---|---|
| $t=0$ | $t=1$ |
| $x=0$ | $x=1$ |
| $v=0$ | $v=0$ |

## 一般变分法

- 目标函数（优化准则）

$$\boxed{x^*(t) = \underset{x(t)}{\operatorname{argmin}} \int_0^T \mathcal{L}\left(x^{(n)}, x^{(n-1)}, \ldots, \dot{x}, x, t\right) dt}$$

- Euler-Lagrange方程

$$\boxed{\frac{\partial \mathcal{L}}{\partial x} - \frac{d}{dt}\left(\frac{\partial \mathcal{L}}{\partial \dot{x}}\right) + \frac{d^2}{dt^2}\left(\frac{\partial \mathcal{L}}{\partial \ddot{x}}\right) + \cdots + (-1)^n \frac{d^n}{dt^n}\left(\frac{\partial \mathcal{L}}{\partial \left(x^{(n)}\right)}\right) = 0}$$

- Lagrange函数

$$\boxed{\mathcal{L}\left(x^{(n)}, x^{(n-1)}, \ldots, \dot{x}, x, t\right) = \left(x^{(n)}\right)^2}$$

- 一些说明
  - $n = 1$ 最小速度
  - $n = 2$ 最小加速度
  - $n = 3$ 最小jerk
  - $n = 4$ 最小snap
  - …

- **MinJerk**

■ 目标函数（优化准则）

$$\boxed{x^*(t) = \underset{x(t)}{\operatorname{argmin}} \int_0^T \mathcal{L}(\dddot{x}, \ddot{x}, \dot{x}, x, t)dt}$$

■ Euler-Lagrange方程

$$\boxed{\frac{\partial \mathcal{L}}{\partial x} - \frac{d}{dt}\left(\frac{\partial \mathcal{L}}{\partial \dot{x}}\right) + \frac{d^2}{dt^2}\left(\frac{\partial \mathcal{L}}{\partial \ddot{x}}\right) - \frac{d^3}{dt^3}\left(\frac{\partial \mathcal{L}}{\partial \dddot{x}}\right) = 0}$$

■ Lagrange函数

$$\boxed{\mathcal{L}\left(x^{(n)}, x^{(n-1)}, \dots, \dot{x}, x, t\right) = (\dddot{x})^2}$$

■ 结果

求解可以得到：$x^{(6)} = 0$，也就是说最优轨迹具有五次多项式的形式！

即：$x(t) = a_5t^5 + a_4t^4 + a_3t^3 + a_2t^2 + a_1t + a_0$

- **一般的变分法**

目标函数(优化准则):

$$x^*(t)=\underset{x(t)}{\operatorname{argmin}}\int_0^T \mathcal{L}\left(x^{(n)},x^{(n-1)},\dots,\dot{x},x,t\right)dt$$

Euler-Lagrange方程:

$$\frac{\partial\mathcal{L}}{\partial x}-\frac{d}{dt}\left(\frac{\partial\mathcal{L}}{\partial\dot{x}}\right)+\frac{d^2}{dt^2}\left(\frac{\partial\mathcal{L}}{\partial\ddot{x}}\right)+\cdots+(-1)^n\frac{d^n}{dt^n}\left(\frac{\partial\mathcal{L}}{\partial\left(x^{(n)}\right)}\right)=0$$

Lagrange函数:

$$\mathcal{L}\left(x^{(n)},x^{(n-1)},\dots,\dot{x},x,t\right)=\left(x^{(n)}\right)^2$$

$$\Downarrow$$

$$x^{(2n)}=0$$

其中:

- $n=1$ 最小速度
- $n=2$ 最小加速度
- $n=3$ 最小jerk
- $n=4$ 最小snap
- …

- **微分方程**

微分方程： $x^{(2n)} = 0$

两类正交多项式：

$$1, t, t^2, t^3, \ldots$$
$$1, \sin t, \cos t, \sin 2t, \cos 2t, \ldots$$

正交多项式：

- 基于Taylor展开的指数多项式（多项式）： $1, t, t^2, t^3, \ldots$
- 基于Fourier展开的三角多项式（三角函数）： $1, \sin t, \cos t, \sin 2t, \cos 2t, \ldots$

基底与拟合：

- 基于多项式基底的轨迹拟合： $1, t, t^2, t^3, \ldots$
- 基于三角函数基底的轨迹拟合： $1, \sin t, \cos t, \sin 2t, \cos 2t, \ldots$

## 轨迹拟合

轨迹拟合：将轨迹形式化一组基底的线性表达，然后根据约束条件求解系数的过程

轨迹拟合的分类：

- 多项式轨迹拟合：
  - 最小加速度： $x(t) = a_0 + a_1 t + a_2 t^2 + a_3 t^3$
  - 最小jerk： $x(t) = a_0 + a_1 t + a_2 t^2 + a_3 t^3 + a_4 t^4 + a_5 t^5$
  - 最小snap： $x(t) = a_0 + a_1 t + a_2 t^2 + a_3 t^3 + a_4 t^4 + a_5 t^5 + a_6 t^6 + a_7 t^7$
- 三角函数轨迹拟合：
  - 正弦轨迹： $x(t) = a_0 + a_1 \cos a_2 t + a_3 \sin a_2 t$
  - 摆线轨迹： $x(t) = a_0 + a_1 t - a_2 \sin a_3 t$
  - Fourier轨迹：

$$x(t) = \frac{A_0}{2} + \sum_{n=1}^{\infty} [A_n \cos nt + B_n \sin nt]$$

$$A_0 = \frac{1}{\pi} \int_{-\pi}^{\pi} x(t) dt$$

$$A_n = \frac{1}{\pi} \int_{-\pi}^{\pi} x(t) \cos nt \, dt$$

$$B_n = \frac{1}{\pi} \int_{-\pi}^{\pi} x(t) \sin nt \, dt$$

## ● 约束条件

为了求解轨迹拟合的系数，需要寻找$(t, x(t))$组合，也就是将约束条件转换为$(t, x(t))$组合

| 轨迹拟合算法 | 形式 | 微分方程 | 约束个数 |
| --- | --- | --- | --- |
| 最小加速度 | $x(t) = a_0 + a_1 t + a_2 t^2 + a_3 t^3$ | $x^{(4)} = 0$ | 4 |
| 最小jerk | $x(t) = a_0 + a_1 t + a_2 t^2 + a_3 t^3 + a_4 t^4 + a_5 t^5$ | $x^{(6)} = 0$ | 6 |
| 最小snap | $x(t) = a_0 + a_1 t + a_2 t^2 + a_3 t^3 + a_4 t^4 + a_5 t^5 + a_6 t^6 + a_7 t^7$ | $x^{(8)} = 0$ | 8 |
| 正弦轨迹 | $x(t) = a_0 + a_1 \cos a_2 t + a_3 \sin a_2 t$ | $x^{(4)} = 0$ | 4 |
| 摆线轨迹 | $x(t) = a_0 + a_1 t - a_2 \sin a_3 t$ | $x^{(4)} = 0$ | 4 |

注意:
- 微分方程阶次与约束个数是相同的
- 多项式次数都是奇次数

- **边界条件例子1**

A cubic path in joint space for the joint variable $q(t)$, or in Cartesian space for a Cartesian coordinate $q(t)$, between two points $q(t_0)$ and $q(t_f)$ is

$$q(t) = a_0 + a_1 t + a_2 t^2 + a_3 t^3 \tag{13.1}$$

四个边界条件:
- 初始位置和初始速度
- 终止位置和终止速度

$$\begin{aligned} q(t_0) &= q_0 & \dot{q}(t_0) &= q'_0 \\ q(t_f) &= q_f & \dot{q}(t_f) &= q'_f. \end{aligned}$$

$$\begin{aligned} q(t) &= a_0 + a_1 t + a_2 t^2 + a_3 t^3 \\ \dot{q}(t) &= a_1 + 2a_2 t + 3a_3 t^2 \end{aligned} \Rightarrow \begin{bmatrix} 1 & t_0 & t_0^2 & t_0^3 \\ 0 & 1 & 2t_0 & 3t_0^2 \\ 1 & t_f & t_f^2 & t_f^3 \\ 0 & 1 & 2t_f & 3t_f^2 \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \end{bmatrix} = \begin{bmatrix} q_0 \\ q'_0 \\ q_f \\ q'_f \end{bmatrix}$$

Jazar R N . Theory of Applied Robotics[J]. 2010.

- **边界条件例子2**

*Consider a harmonic path between two points* $q(t_0)$ *and* $q(t_f)$

$$q(t) = a_0 + a_1 \cos a_2 t + a_3 \sin a_2 t \tag{13.129}$$

四个边界条件:

- 初始位置和初始速度
- 终止位置和终止速度

$$\begin{aligned} q(t_0) &= q_0 \qquad \dot{q}(t_0) = 0 \\ q(t_f) &= q_f \qquad \dot{q}(t_f) = 0 \end{aligned}$$

$$q(t) = \frac{1}{2}\left(q_f + q_0 - (q_f - q_0)\cos\frac{\pi(t - t_0)}{t_f - t_0}\right)$$

Jazar R N . Theory of Applied Robotics[J]. 2010.

- **简化问题——time shift**

为了简化求解过程的计算量，我们通常会采用两个方法：
（1）time shift
（2）rest-to-rest

A cubic path in joint space for the joint variable $q(t)$, or in Cartesian space for a Cartesian coordinate $q(t)$, between two points $q(t_0)$ and $q(t_f)$ is

$$q(t) = a_0 + a_1 t + a_2 t^2 + a_3 t^3 \tag{13.1}$$

$$\begin{aligned} q(t_0) &= q_0 & \dot{q}(t_0) = 0 \\ q(t_f) &= q_f & \dot{q}(t_f) = 0 \end{aligned}$$

time shift：

$$q(t) = a_0 + a_1 (t - t_0) + a_2 t (t - t_0)^2 + a_3 (t - t_0)^3$$

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & (t_f - t_0) & (t_f - t_0)^2 & (t_f - t_0)^3 \\ 0 & 1 & 2(t_f - t_0) & 3(t_f - t_0)^2 \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \end{bmatrix} = \begin{bmatrix} q_0 \\ q_0' \\ q_f \\ q_f' \end{bmatrix}$$

换言之：$t_0 = 0$

Jazar R N . Theory of Applied Robotics[J]. 2010.

- **简化问题——rest-to-rest**

rest-to-rest：仅考虑位置约束，速度即速度的多阶导数（如果需要）在边界条件处为零

*A five degree polynomial can satisfy these conditions*

$$q(t) = a_0 + a_1 t + a_2 t^2 + a_3 t^3 + a_4 t^4 + a_5 t^5 \qquad (13.54)$$

完整边界条件与rest-to-rest边界条件

$$\begin{aligned} q(t_0) &= q_0 \qquad \dot{q}(t_0) = q_0' \qquad \ddot{q}(t_0) = q_0'' \\ q(t_f) &= q_f \qquad \dot{q}(t_f) = q_f' \qquad \ddot{q}(t_f) = q_f'' \end{aligned} \qquad \begin{aligned} q(0) &= 10\,\text{deg} \qquad \dot{q}(0) = 0 \qquad \ddot{q}(0) = 0 \\ q(1) &= 45\,\text{deg} \qquad \dot{q}(1) = 0 \qquad \ddot{q}(1) = 0 \end{aligned}$$

完整约束求解与rest-to-rest约束求解

$$\begin{bmatrix} 1 & t_0 & t_0^2 & t_0^3 & t_0^4 & t_0^5 \\ 0 & 1 & 2t_0 & 3t_0^2 & 4t_0^3 & 5t_0^4 \\ 0 & 0 & 2 & 6t_0 & 12t_0^2 & 20t_0^3 \\ 1 & t_f & t_f^2 & t_f^3 & t_f^4 & t_f^5 \\ 0 & 1 & 2t_f & 3t_f^2 & 4t_f^3 & 5t_f^4 \\ 0 & 0 & 2 & 6t_f & 12t_f^2 & 20t_f^3 \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \\ a_4 \\ a_5 \end{bmatrix} = \begin{bmatrix} q_0 \\ q_0' \\ q_0'' \\ q_f \\ q_f' \\ q_f'' \end{bmatrix} \qquad \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 2 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 2 & 3 & 4 & 5 \\ 0 & 0 & 2 & 6 & 12 & 20 \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \\ a_4 \\ a_5 \end{bmatrix} = \begin{bmatrix} 10 \\ 0 \\ 0 \\ 45 \\ 0 \\ 0 \end{bmatrix}$$

Jazar R N . Theory of Applied Robotics[J]. 2010.

## 时间尺度——从路径到轨迹

路径：路径规划通常会在构型空间生成一条二阶连续可微的曲线，即$Q, s: [0,1] \rightarrow q(s)$

时间尺度：标量函数：$f, t: [0, t_f] \rightarrow s(t)$: $[0,1]$

轨迹：为路径赋予时间尺度，即：$Qf, t: [0, t_f] \rightarrow q\big(s(t)\big)$

A cubic path in joint space for the joint variable $q(t)$, or in Cartesian space for a Cartesian coordinate $q(t)$, between two points $q(t_0)$ and $q(t_f)$ is

$$q(t) = a_0 + a_1 t + a_2 t^2 + a_3 t^3 \tag{13.1}$$

$$\begin{aligned} q(t_0) &= q_0 \qquad \dot{q}(t_0) = 0 \\ q(t_f) &= q_f \qquad \dot{q}(t_f) = 0 \end{aligned}$$

路径函数：$q(s) = q_0 + s\big(q_f - q_0\big)$

时间尺度：$s(t) = a_0 + a_1 t + a_2 t^2 + a_3 t^3$

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & t_f & t_f^2 & t_f^3 \\ 0 & 1 & 2t_f & 3t_f^2 \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{bmatrix}$$

注意：

- 时间尺度连接了路径规划与轨迹优化
- 时间尺度在“时间最优”轨迹优化中使用较多，这里只是引入了一下理论

Choset. Principles of Robot Motion: Theory, Algorithms, and Implementations[J]. 2005.

- **轨迹优化问题**

## Trajectory Optimization Problem*

** Given our set of assumptions*

$$\underbrace{\min_{t_0, t_F, \boldsymbol{x}(t), \boldsymbol{u}(t)}}_{\text{functions!}} \quad \underbrace{J\big(t_0, t_F, \boldsymbol{x}(t_0), \boldsymbol{x}(t_F)\big)}_{\text{boundary objective function}} + \underbrace{\int_{t_0}^{t_F} w\big(\tau, \boldsymbol{x}(\tau), \boldsymbol{u}(\tau)\big)\, d\tau}_{\text{integral objective function}}$$

calculus!

subject to:

$$\boldsymbol{g}\big(t_0, t_F, \boldsymbol{x}(t_0), \boldsymbol{x}(t_F)\big) \leq \boldsymbol{0}$$
boundary constraints

$$t_{\text{low}} \leq t_0 < t_F \leq t_{\text{upp}}$$
bound on initial and final time

$$\boldsymbol{x}_{0,\text{low}} \leq \boldsymbol{x}(t_0) \leq \boldsymbol{x}_{0,\text{upp}}$$
bound on initial state

$$\boldsymbol{x}_{F,\text{low}} \leq \boldsymbol{x}(t_F) \leq \boldsymbol{x}_{F,\text{upp}}$$
bound on final state

$$\dot{\boldsymbol{x}}(t) = \boldsymbol{f}\big(t, \boldsymbol{x}(t), \boldsymbol{u}(t)\big)$$
continuous dynamics

$$\boldsymbol{h}\big(t, \boldsymbol{x}(t), \boldsymbol{u}(t)\big) \leq \boldsymbol{0}$$
path constraints

$$\boldsymbol{x}_{\text{low}} \leq \boldsymbol{x}(t) \leq \boldsymbol{x}_{\text{upp}}$$
continuous bounds on state

$$\boldsymbol{u}_{\text{low}} \leq \boldsymbol{u}(t) \leq \boldsymbol{u}_{\text{upp}}$$
continuous bounds on control

两种求解思路:
- ➢ 解析法（变分法）
- ➢ 数值法

http://www.matthewpeterkelly.com/tutorials/trajectoryOptimization/cartPoleCollocation.svg

- **数值积分问题**

数值积分公式的一般形式：

$$I_n(f)=\sum_{k=0}^{n}A_k f(x_k)\approx\int_a^b f(x)dx$$

其中，

求积结点：$a\leq x_0<x_1<\cdots<x_n\leq b$

求积系数：$A_k(k=0,1,2,\ldots n)$仅与结点有关

插值公式的一般形式：

$$f(x)\approx\sum_{k=0}^{n}l_k(x)f(x_k)$$

其中，

插值结点：$a\leq x_0<x_1<\cdots<x_n\leq b$

插值型数值积分公式的一般形式：

$$\int_a^b f(x)dx\approx\int_a^b\sum_{k=0}^{n}l_k(x)f(x_k)\,dx$$

$$=\sum_{k=0}^{n}f(x_k)\int_a^b l_k(x)dx=I_n(f)$$

其中，

求积结点：$a\leq x_0<x_1<\cdots<x_n\leq b$

求积系数：$A_k=\int_a^b l_k(x)dx$

具体算法：

梯形公式：$I_1(f)=\frac{b-a}{2}\left(f(a)+f(b)\right)$

Simspon公式：$I_2(f)=\frac{b-a}{6}\left(f(a)+4f\left(\frac{a+b}{2}\right)+f(b)\right)$

Cote公式 / 复化梯形公式 / 复化Simspon公式

Gauss求积公式

李庆扬. 数值分析[M]. 清华大学出版社, 2001.

# 03 从数值计算到轨迹插值



## ● 从最优到NLP

### Transcription

**trajectory optimization problem** → **constrained parameter optimization**

trajectory optimization problem:

- decision variables are vector ***functions***
- infinite dimensional
- ***differential*** equations

constrained parameter optimization:

- decision variables are real ***numbers***
- finite dimensional
- ***algebraic*** equations

$$\underbrace{\min_{t_0, t_F, \boldsymbol{x}(t), \boldsymbol{u}(t)}}_{\text{functions!}} \underbrace{J\big(t_0, t_F, \boldsymbol{x}(t_0), \boldsymbol{x}(t_F)\big)}_{\text{boundary objective function}} + \underbrace{\int_{t_0}^{t_F} w\big(\tau, \boldsymbol{x}(\tau), \boldsymbol{u}(\tau)\big)\, d\tau}_{\text{integral objective function}}$$

calculus!

subject to:

$$\underbrace{\boldsymbol{g}\big(t_0, t_F, \boldsymbol{x}(t_0), \boldsymbol{x}(t_F)\big) \le \boldsymbol{0}}_{\text{boundary constraints}} \qquad \underbrace{\dot{\boldsymbol{x}}(t) = \boldsymbol{f}\big(t, \boldsymbol{x}(t), \boldsymbol{u}(t)\big)}_{\text{continuous dynamics}}$$

$$\underbrace{t_{\text{low}} \le t_0 < t_F \le t_{\text{upp}}}_{\text{bound on initial and final time}} \qquad \underbrace{\boldsymbol{h}\big(t, \boldsymbol{x}(t), \boldsymbol{u}(t)\big) \le \boldsymbol{0}}_{\text{path constraints}}$$

$$\underbrace{\boldsymbol{x}_{0,\text{low}} \le \boldsymbol{x}(t_0) \le \boldsymbol{x}_{0,\text{upp}}}_{\text{bound on initial state}} \qquad \underbrace{\boldsymbol{x}_{\text{low}} \le \boldsymbol{x}(t) \le \boldsymbol{x}_{\text{upp}}}_{\text{continuous bounds on state}}$$

$$\underbrace{\boldsymbol{x}_{F,\text{low}} \le \boldsymbol{x}(t_F) \le \boldsymbol{x}_{F,\text{upp}}}_{\text{bound on final state}} \qquad \underbrace{\boldsymbol{u}_{\text{low}} \le \boldsymbol{u}(t) \le \boldsymbol{u}_{\text{upp}}}_{\text{continuous bounds on control}}$$

→

**Non-Linear Program (NLP)**

$$\min_{\boldsymbol{z}} f(\boldsymbol{z}) \qquad \text{subject to:}$$

$$\boldsymbol{g}(\boldsymbol{z}) \le \boldsymbol{0}$$

$$\boldsymbol{h}(\boldsymbol{z}) = \boldsymbol{0}$$

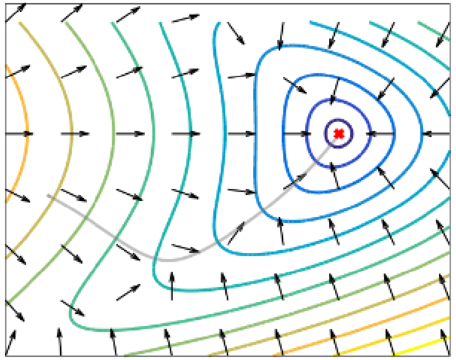
$$\boldsymbol{z}_{\text{low}} \le \boldsymbol{z} \le \boldsymbol{z}_{\text{upp}}$$

Transcription思路

- ➢ 原始问题：泛函优化$\boldsymbol{u}(\boldsymbol{x})$
- ➢ 新问题：函数优化$\boldsymbol{u}(t)$

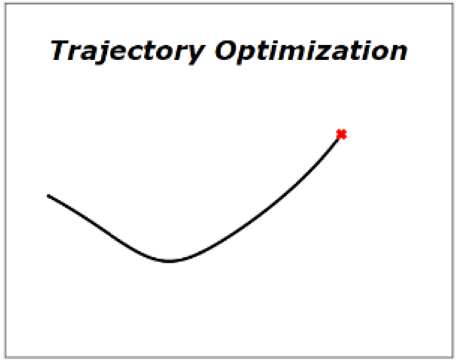
Closed-Loop Solution

$\boldsymbol{u} = \boldsymbol{u}(\boldsymbol{x})$



Open-Loop Solution

$\boldsymbol{u} = \boldsymbol{u}(t)$




http://www.matthewpeterkelly.com/tutorials/trajectoryOptimization/cartPoleCollocation.svg

## ● 数值计算与直接配置法(Direct Collocation)

| | |
|---|---|
| $t_k$ | time at knot point $k$ |
| $N$ | number of trajectory (spline) segments |
| $h_k = t_{k+1} - t_k$ | duration of spline segment $k$ |
| $\boldsymbol{x}_k = \boldsymbol{x}(t_k)$ | state at knot point $k$ |
| $\boldsymbol{u}_k = \boldsymbol{u}(t_k)$ | control at knot point $k$ |
| $w_k = w(t_k, \boldsymbol{x}_k, \boldsymbol{u}_k)$ | integrand of objective function at knot point $k$ |
| $\boldsymbol{f}_k = \boldsymbol{f}(t_k, \boldsymbol{x}_k, \boldsymbol{u}_k)$ | system dynamics at knot point $k$ |
| $\dot{q} = \frac{d}{dt}q, \quad \ddot{q} = \frac{d^2}{dt^2}q$ | first and second time-derivatives of $q$ |

配置点:

- ■ 连续时间的离散化 $t \to t_0, \dots, t_k, \dots, t_N$
- ■ 连续状态的离散化 $\boldsymbol{x} \to \boldsymbol{x}_0, \dots, \boldsymbol{x}_k, \dots, \boldsymbol{x}_N$
- ■ 连续控制的离散化 $\boldsymbol{u} \to \boldsymbol{u}_0, \dots, \boldsymbol{u}_k, \dots, \boldsymbol{u}_N$

配置约束:

- ■ 系统动力学约束 $\dot{\boldsymbol{x}} = \boldsymbol{f}(t, \boldsymbol{x}, \boldsymbol{u}) \to \boldsymbol{x}_{k+1} = \boldsymbol{f}_k(\boldsymbol{x}_k, \boldsymbol{u}_k)$
- ■ 边界条件约束 $\boldsymbol{x}(t_0), \boldsymbol{x}(t_f) \to \boldsymbol{x}_0, \boldsymbol{x}_N$

配置目标函数:

- ■ 目标函数 $\int_{t_0}^{t_F} w(\tau, \boldsymbol{x}(\tau), \boldsymbol{u}(\tau))d\tau \to \sum_{k=0}^{N} w_k(t_k, \boldsymbol{x}_k, \boldsymbol{u}_k)$

具体算法:

梯形公式: $I_1(f) = \frac{b-a}{2}\left(f(a) + f(b)\right)$

Simspon公式: $I_2(f) = \frac{b-a}{6}\left(f(a) + 4f\left(\frac{a+b}{2}\right) + f(b)\right)$

Kelly M. An introduction to trajectory optimization: How to do your own direct collocation[J]. SIAM

# 03 从数值计算到轨迹插值



- **梯形配置法(Trapezoidal Collocation)问题**

梯形公式：$I_1(f) = \frac{b-a}{2}\big(f(a) + f(b)\big)$

■ 目标函数

➢ 连续积分变为离散求和$\int w(\cdot) \to \sum c_k w_k$

➢ 所有函数值变为配置点函数值$w(t_k) = w_k$

$$\boxed{\int_{t_0}^{t_F} w\big(\tau, \boldsymbol{x}(\tau), \boldsymbol{u}(\tau)\big)\, d\tau \quad \approx \quad \sum_{k=0}^{N-1} \tfrac{1}{2} h_k \cdot \big(w_k + w_{k+1}\big)}$$

$$h_k = t_{k+1} - t_k$$

■ 系统动力学

➢ 状态方程的梯形积分

➢ 连续动力学变为配置点动力学

$$\boxed{\boldsymbol{x}_{k+1} - \boldsymbol{x}_k = \tfrac{1}{2}\, h_k \cdot \big(\boldsymbol{f}_{k+1} + \boldsymbol{f}_k\big)}$$

$$\dot{\boldsymbol{x}} = \boldsymbol{f},$$

$$\int_{t_k}^{t_{k+1}} \dot{\boldsymbol{x}}\, dt = \int_{t_k}^{t_{k+1}} \boldsymbol{f}\, dt,$$

$$\boldsymbol{x}_{k+1} - \boldsymbol{x}_k \approx \tfrac{1}{2}\, h_k \cdot \big(\boldsymbol{f}_{k+1} + \boldsymbol{f}_k\big).$$

■ 约束

➢ 状态和控制约束离散化

➢ 路径约束离散化

➢ 边界约束变为初始配置点

$$\begin{array}{lcl}
\boldsymbol{x} < \boldsymbol{0} & \rightarrow & \boldsymbol{x}_k < \boldsymbol{0} \quad \forall k, \\
\boldsymbol{u} < \boldsymbol{0} & \rightarrow & \boldsymbol{u}_k < \boldsymbol{0} \quad \forall k. \\
\boldsymbol{g}(t, \boldsymbol{x}, \boldsymbol{u}) < \boldsymbol{0} & \rightarrow & \boldsymbol{g}(t_k, \boldsymbol{x}_k, \boldsymbol{u}_k) < \boldsymbol{0} \quad \forall k. \\
\boldsymbol{h}\big(t_0, \boldsymbol{x}(t_0), \boldsymbol{u}(t_0)\big) < \boldsymbol{0} & \rightarrow & \boldsymbol{h}\big(t_0, \boldsymbol{x}_0, \boldsymbol{u}_0\big) < \boldsymbol{0}.
\end{array}$$

Kelly M. An introduction to trajectory optimization: How to do your own direct collocation[J]. SIAM

- **梯形配置法(Trapezoidal Collocation)求解**

梯形公式：$I_1(f) = \frac{b-a}{2}\big(f(a) + f(b)\big)$

■ 控制函数

- 控制函数采用梯形配置进行求解
- 梯形配置使用分段线性多项式（线性插值）
- 线性插值结点跟配置点数量相同

$$\boldsymbol{u}(t) \approx \boldsymbol{u}_k + \frac{\tau}{h_k}\left(\boldsymbol{u}_{k+1} - \boldsymbol{u}_k\right)$$

$$\tau = t - t_k$$

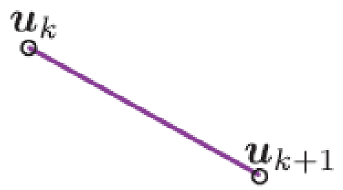
$$h_k = t_{k+1} - t_k$$

■ 系统动力学

- 假设动力学方程$\dot{\boldsymbol{x}} = \boldsymbol{f}(t, \boldsymbol{x}, \boldsymbol{u})$与$\boldsymbol{u}$是线性关系
- $\boldsymbol{f}$也要采用梯形配置进行求解
- 动力学方程使用线性插值

$$\boldsymbol{f}(t) = \dot{\boldsymbol{x}}(t) \approx \boldsymbol{f}_k + \frac{\tau}{h_k}\left(\boldsymbol{f}_{k+1} - \boldsymbol{f}_k\right)$$

control:
linear spline



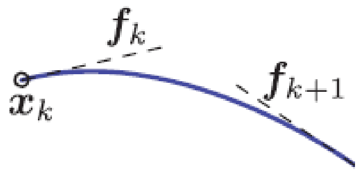

■ 状态

- 状态从系统动力学积分得到
- 状态求解使用二次多项式（二次插值）

$$\boldsymbol{x}(t) = \int \dot{\boldsymbol{x}}(t)\, d\tau \approx \boldsymbol{c} + \boldsymbol{f}_k \tau + \frac{\tau^2}{2h_k}\left(\boldsymbol{f}_{k+1} - \boldsymbol{f}_k\right)$$

$$\boldsymbol{x}(t) \approx \boldsymbol{x}_k + \boldsymbol{f}_k \tau + \frac{\tau^2}{2h_k}\left(\boldsymbol{f}_{k+1} - \boldsymbol{f}_k\right)$$

state:
quadratic spline




Kelly M. An introduction to trajectory optimization: How to do your own direct collocation[J]. SIAM

## 03 从数值计算到轨迹插值



- **Simpson配置法(Hermite–Simpson Collocation)问题**

Simpson公式：$I_2(f)=\frac{b-a}{6}\left(f(a)+4f\left(\frac{a+b}{2}\right)+f(b)\right)$

■ 目标函数

- 连续积分变为离散求和$\int w(\cdot)\to\sum c_k w_k$
- 所有函数值变为配置点函数值$w(t_k)=w_k$
- 多了一项$k+\frac{1}{2}$

$$\int_{t_0}^{t_F} w(\tau)\,d\tau \approx \sum_{k=0}^{N-1}\frac{h_k}{6}\left(w_k+4w_{k+\frac{1}{2}}+w_{k+1}\right).$$

$$h_k=t_{k+1}-t_k$$

■ 系统动力学

- 状态方程的Simpson积分
- 连续动力学变为配置点动力学

$$x_{k+1}-x_k=\frac{1}{6}h_k\left(f_k+4f_{k+\frac{1}{2}}+f_{k+1}\right).$$

$$x_{k+\frac{1}{2}}=\frac{1}{2}\left(x_k+x_{k+1}\right)+\frac{h_k}{8}\left(f_k-f_{k+1}\right).$$

$$\dot{x}=f,$$

$$\int_{t_k}^{t_{k+1}}\dot{x}\,dt=\int_{t_k}^{t_{k+1}} f\,dt.$$

■ 约束

- 状态和控制约束离散化
- 路径约束离散化
- 边界约束变为初始配置点

$$x<0 \quad\to\quad \begin{aligned} x_k &< 0,\\ x_{k+\frac{1}{2}} &< 0, \end{aligned}$$

$$u<0 \quad\to\quad \begin{aligned} u_k &< 0,\\ u_{k+\frac{1}{2}} &< 0. \end{aligned}$$

$$g(t,x,u)<0 \quad\to\quad \begin{aligned} g(t_k,x_k,u_k) &< 0,\\ g(t_{k+\frac{1}{2}},x_{k+\frac{1}{2}},u_{k+\frac{1}{2}}) &< 0. \end{aligned}$$

$$h(t_0,x(t_0),u(t_0))<0 \quad\to\quad h(t_0,x_0,u_0)<0.$$

Kelly M. An introduction to trajectory optimization: How to do your own direct collocation[J]. SIAM

- **Simpson配置法(Hermite–Simpson Collocation)求解**

Simpson公式：$I_2(f) = \frac{b-a}{6}\left(f(a) + 4f\left(\frac{a+b}{2}\right) + f(b)\right)$

■ 控制函数

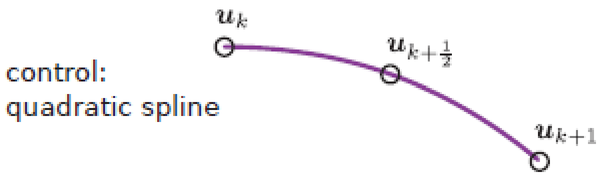
➢ 控制函数采用Simpson配置进行求解

➢ 梯形配置使用分段二次多项式（二次插值）

➢ 线性插值结点跟配置点数量不同

$$\tau = t - t_k$$

$$h_k = t_{k+1} - t_k$$



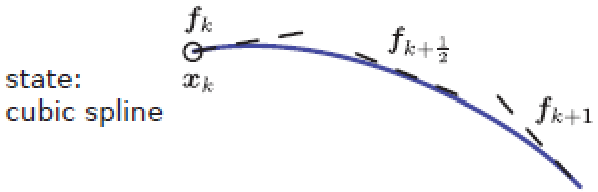

$$\boxed{u(t) = \frac{2}{h_k^2}\left(\tau - \frac{h_k}{2}\right)(\tau - h_k)u_k - \frac{4}{h_k^2}(\tau)(\tau - h_k)u_{k+\frac{1}{2}} + \frac{2}{h_k^2}(\tau)\left(\tau - \frac{h_k}{2}\right)u_{k+1}.}$$

■ 系统动力学

➢ 假设动力学方程$\dot{\boldsymbol{x}} = \boldsymbol{f}(t, \boldsymbol{x}, \boldsymbol{u})$与$\boldsymbol{u}$是线性关系

➢ $\boldsymbol{f}$也要采用Simpson配置进行求解

➢ 动力学方程使用二次插值




$$\boxed{f(t) = \dot{x} = \frac{2}{h_k^2}\left(\tau - \frac{h_k}{2}\right)(\tau - h_k)f_k - \frac{4}{h_k^2}(\tau)(\tau - h_k)f_{k+\frac{1}{2}} + \frac{2}{h_k^2}(\tau)\left(\tau - \frac{h_k}{2}\right)f_{k+1}.}$$

■ 状态

➢ 状态从系统动力学积分得到

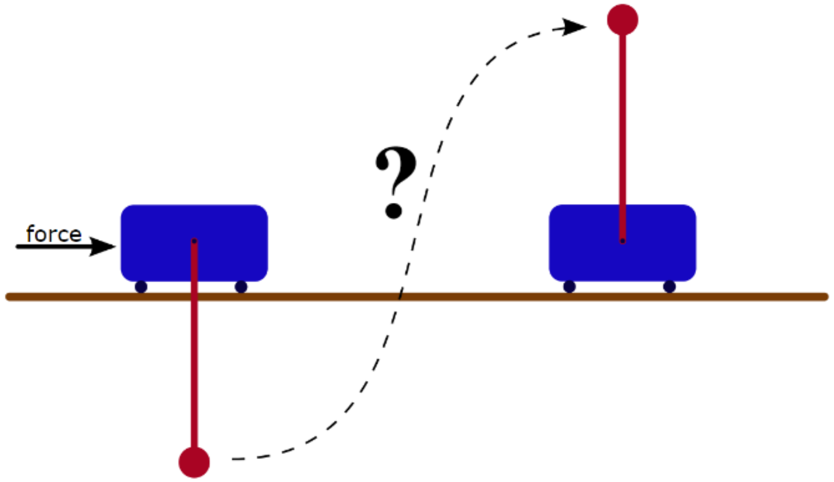
➢ 状态求解使用三次多项式（三次插值）

$$\boxed{\begin{aligned} x(t) = x_k + f_k\left(\frac{\tau}{h_k}\right) + \frac{1}{2}\left(-3f_k + 4f_{k+\frac{1}{2}} - f_{k+1}\right)\left(\frac{\tau}{h_k}\right)^2 \\ + \frac{1}{3}\left(2f_k - 4f_{k+\frac{1}{2}} + 2f_{k+1}\right)\left(\frac{\tau}{h_k}\right)^3. \end{aligned}}$$

- **配置法例子**

■ 小车倒立摆模型

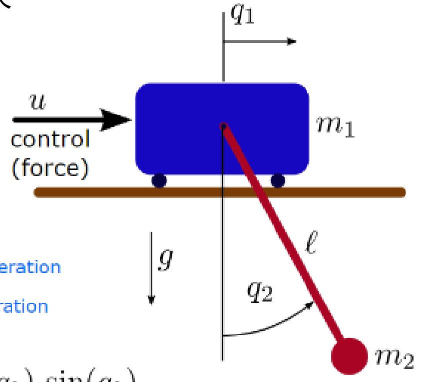
- 问题表述
- 系统建模
- 轨迹优化问题形式化




### 系统建模

$$\underset{\text{state}}{\boldsymbol{x}} = \begin{bmatrix} q_1 \\ q_2 \\ \dot{q}_1 \\ \dot{q}_2 \end{bmatrix} \begin{array}{l} \text{cart horizontal position} \\ \text{pole angle} \\ \text{cart horizontal velocity} \\ \text{pole angular rate} \end{array}$$

$$\underset{\text{dynamics}}{\dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x}, u)} = \begin{bmatrix} \dot{q}_1 \\ \dot{q}_2 \\ \ddot{q}_1 \\ \ddot{q}_2 \end{bmatrix} \begin{array}{l} \\ \\ \text{cart horizontal acceleration} \\ \text{pole angular acceleration} \end{array}$$




$$\ddot{q}_1 = \frac{\ell\, m_2 \sin(q_2)\, \dot{q}_2^2 + u + m_2\, g\, \cos(q_2)\, \sin(q_2)}{m_1 + m_2\left(1 - \cos^2(q_2)\right)}$$

$$\ddot{q}_2 = -\frac{\ell\, m_2 \cos(q_2)\, \sin(q_2)\, \dot{q}_2^2 + u\, \cos(q_2) + (m_1 + m_2)\, g\, \sin(q_2)}{\ell\, m_1 + \ell m_2\left(1 - \cos^2(q_2)\right)}$$

### 轨迹优化问题形式化

$$\boxed{\min_{\boldsymbol{x}(t), u(t)} \int_0^T u^2(\tau)\, d\tau}$$

subject to:

Dynamics

$$\boxed{\dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x}, u)}$$

Path Bounds

| | |
|---|---|
| $-d_{\max} \le q_1 \le d_{\max}$ | keep cart on track |
| $-u_{\max} \le u \le u_{\max}$ | motor saturation |

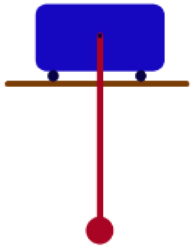
Parameters

| | |
|---|---|
| $d$ | horizontal travel |
| $d_{\max}$ | track half length |
| $u_{\max}$ | peak motor force |
| $T$ | trajectory duration |

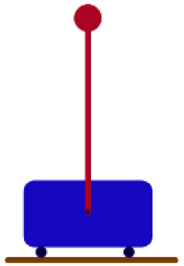
Bounds on initial and final state

initial state

$q_1(0) = 0$
$q_2(0) = 0$
$\dot{q}_1(0) = 0$
$\dot{q}_2(0) = 0$



final state



$q_1(T) = d$
$q_2(T) = \pi$
$\dot{q}_1(T) = 0$
$\dot{q}_2(T) = 0$

Kelly M. An introduction to trajectory optimization: How to do your own direct collocation[J]. SIAM

- **配置法例子**
  - 梯形配置

minimize

(6.4)

$$J = \sum_{k=0}^{N-1} \frac{h_k}{2}\left(u_k^2 + u_{k+1}^2\right), \qquad \textbf{objective function,}$$

with decision variables

(6.5)

$$x_0, \ldots, x_N \quad u_0, \ldots, u_N,$$

subject to

(6.6)

$$\frac{1}{2}h_k\left(\boldsymbol{f}_{k+1} + \boldsymbol{f}_k\right) = x_{k+1} - x_k, \qquad k \in 0, \ldots, (N-1), \qquad \textbf{collocation constraints,}$$

(6.7)

$$-d_{\max} \le q_1 \le d_{\max}, \qquad \textbf{path constraints,}$$

(6.8)

$$-u_{\max} \le u \le u_{\max}, \qquad \textbf{path constraints,}$$

(6.9)

$$x_0 = \boldsymbol{0}, \quad x_N = [d, \pi, 0, 0]^T, \qquad \textbf{boundary constraints.}$$

Note that $h_k = t_{k+1} - t_k$. Here, we will use a uniform grid, so $t_k = k\frac{T}{N}$, where $N$ is the number of segments used in the transcription. In general, you could solve this problem on an arbitrary grid; in other words, each $h_k$ could be different.

- **配置法例子**
  - Simpson配置

minimize

(6.10)

$$J = \sum_{k=0}^{N-1} \frac{h_k}{6}\left(u_k^2 + 4u_{k+\frac{1}{2}}^2 + u_{k+1}^2\right),$$ objective function,

with decision variables

$$x_0,\ x_{0+\frac{1}{2}}, \ldots, x_N, \qquad u_0,\ u_{0+\frac{1}{2}}, \ldots, u_N,$$

subject to

(6.11)

$$x_{k+\frac{1}{2}} = \frac{1}{2}\left(x_k + x_{k+1}\right) + \frac{h_k}{8}\left(\boldsymbol{f}_k - \boldsymbol{f}_{k+1}\right), \qquad k \in 0, \ldots, (N-1),$$ interpolation constraints,

(6.12)

$$\frac{h_k}{6}\left(\boldsymbol{f}_k + 4\boldsymbol{f}_{k+\frac{1}{2}} + \boldsymbol{f}_{k+1}\right) = x_{k+1} - x_k, \qquad k \in 0, \ldots, (N-1),$$ collocation constraints,

(6.13)

$$-d_{\max} \leq q_1 \leq d_{\max},$$ path constraints,

(6.14)

$$-u_{\max} \leq u \leq u_{\max},$$ path constraints,

(6.15)

$$x_0 = \boldsymbol{0}, \quad x_N = [d, \pi, 0, 0]^T,$$ boundary constraints.

# 03 从数值计算到轨迹插值



- **配置法例子**

■ 梯形配置

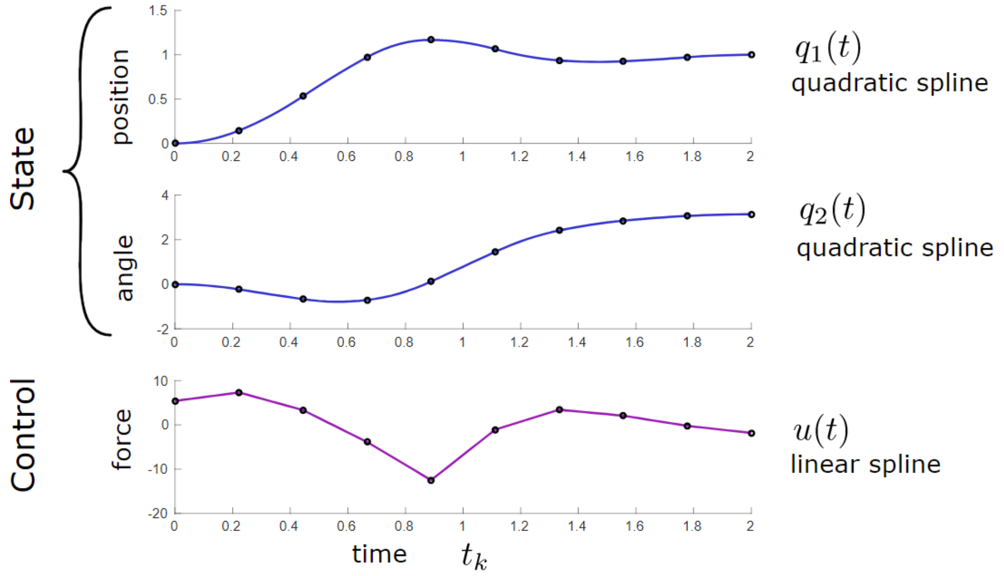
➢ 控制函数为线性插值

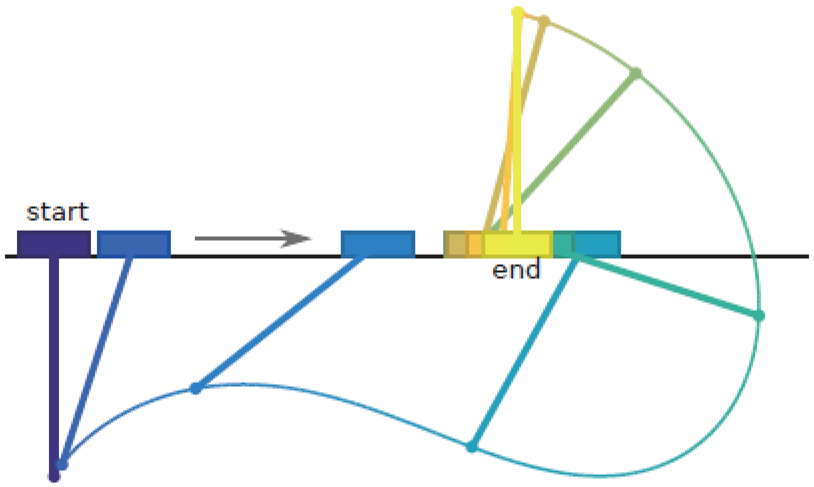
➢ 状态函数为二次插值

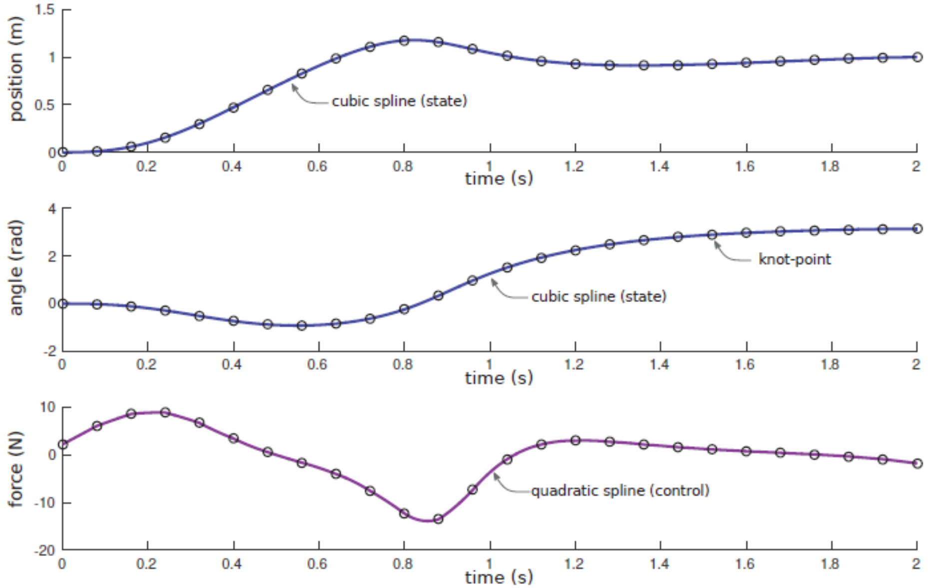
■ Simpson配置

➢ 控制函数为二次插值

➢ 状态函数为三次插值



# 04 轨迹拟合 & 轨迹插值



## ● 轨迹优化问题

### Trajectory Optimization Problem*

* ***Given our set of assumptions***

$$\underbrace{\min_{t_0, t_F, \boldsymbol{x}(t), \boldsymbol{u}(t)}}_{\text{functions!}} \quad J\big(t_0, t_F, \boldsymbol{x}(t_0), \boldsymbol{x}(t_F)\big) + \int_{t_0}^{t_F} w\big(\tau, \boldsymbol{x}(\tau), \boldsymbol{u}(\tau)\big)\, d\tau$$

boundary objective function　　integral objective function

calculus!

subject to:

$$\dot{\boldsymbol{x}}(t) = \boldsymbol{f}\big(t, \boldsymbol{x}(t), \boldsymbol{u}(t)\big)$$

continuous dynamics

$$\boldsymbol{g}\big(t_0, t_F, \boldsymbol{x}(t_0), \boldsymbol{x}(t_F)\big) \le \boldsymbol{0}$$

boundary constraints

$$\boldsymbol{h}\big(t, \boldsymbol{x}(t), \boldsymbol{u}(t)\big) \le \boldsymbol{0}$$

path constraints

$$t_{\text{low}} \le t_0 < t_F \le t_{\text{upp}}$$

bound on initial and final time

$$\boldsymbol{x}_{0,\text{low}} \le \boldsymbol{x}(t_0) \le \boldsymbol{x}_{0,\text{upp}}$$

bound on initial state

$$\boldsymbol{x}_{\text{low}} \le \boldsymbol{x}(t) \le \boldsymbol{x}_{\text{upp}}$$

continuous bounds on state

$$\boldsymbol{x}_{F,\text{low}} \le \boldsymbol{x}(t_F) \le \boldsymbol{x}_{F,\text{upp}}$$

bound on final state

$$\boldsymbol{u}_{\text{low}} \le \boldsymbol{u}(t) \le \boldsymbol{u}_{\text{upp}}$$

continuous bounds on control

两种求解思路:

- ➢ 解析法（变分法）
- ➢ 数值法

http://www.matthewpeterkelly.com/tutorials/trajectoryOptimization/cartPoleCollocation.svg

- **轨迹优化求解**

**Transcription**

trajectory optimization problem → constrained parameter optimization

- decision variables are vector ***functions***
- infinite dimensional
- ***differential*** equations

- decision variables are real ***numbers***
- finite dimensional
- ***algebraic*** equations

$$\underbrace{\min_{t_0, t_F, \boldsymbol{x}(t), \boldsymbol{u}(t)}}_{\text{functions!}} \underbrace{J\big(t_0, t_F, \boldsymbol{x}(t_0), \boldsymbol{x}(t_F)\big)}_{\text{boundary objective function}} + \int_{t_0}^{t_F} \underbrace{w\big(\tau, \boldsymbol{x}(\tau), \boldsymbol{u}(\tau)\big)\, d\tau}_{\text{integral objective function}}$$

calculus!

subject to:

$\boldsymbol{g}\big(t_0, t_F, \boldsymbol{x}(t_0), \boldsymbol{x}(t_F)\big) \le \boldsymbol{0}$ boundary constraints

$\dot{\boldsymbol{x}}(t) = \boldsymbol{f}\big(t, \boldsymbol{x}(t), \boldsymbol{u}(t)\big)$ continuous dynamics

$t_{\text{low}} \le t_0 < t_F \le t_{\text{upp}}$ bound on initial and final time

$\boldsymbol{h}\big(t, \boldsymbol{x}(t), \boldsymbol{u}(t)\big) \le \boldsymbol{0}$ path constraints

$\boldsymbol{x}_{0,\text{low}} \le \boldsymbol{x}(t_0) \le \boldsymbol{x}_{0,\text{upp}}$ bound on initial state

$\boldsymbol{x}_{\text{low}} \le \boldsymbol{x}(t) \le \boldsymbol{x}_{\text{upp}}$ continuous bounds on state

$\boldsymbol{x}_{F,\text{low}} \le \boldsymbol{x}(t_F) \le \boldsymbol{x}_{F,\text{upp}}$ bound on final state

$\boldsymbol{u}_{\text{low}} \le \boldsymbol{u}(t) \le \boldsymbol{u}_{\text{upp}}$ continuous bounds on control

→

**Non-Linear Program (NLP)**

$$\min_{\boldsymbol{z}} f(\boldsymbol{z}) \qquad \text{subject to:}$$

$$\boldsymbol{g}(\boldsymbol{z}) \le \boldsymbol{0}$$

$$\boldsymbol{h}(\boldsymbol{z}) = \boldsymbol{0}$$

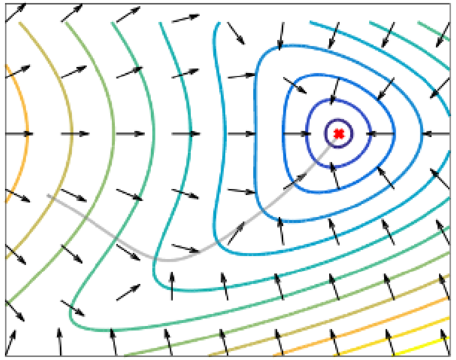
$$\boldsymbol{z}_{\text{low}} \le \boldsymbol{z} \le \boldsymbol{z}_{\text{upp}}$$

两种思路

- 连续问题：泛函优化$\boldsymbol{u}(\boldsymbol{x})$
- 离散问题：函数优化$\boldsymbol{u}(t)$

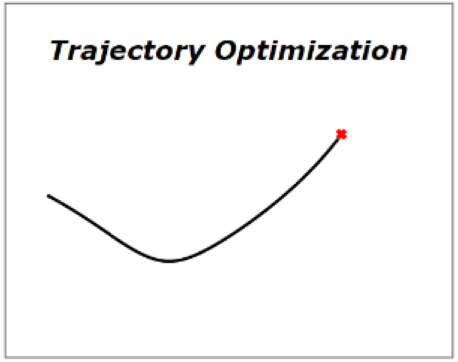
Closed-Loop Solution

$\boldsymbol{u} = \boldsymbol{u}(\boldsymbol{x})$



Open-Loop Solution

$\boldsymbol{u} = \boldsymbol{u}(t)$




http://www.matthewpeterkelly.com/tutorials/trajectoryOptimization/cartPoleCollocation.svg

- **泛函优化与微分方程**

■ 无约束条件泛函极值

➢ 变分法与Euler-Lagrange方程

目标函数

$$\mathcal{J}=\int_{t_0}^{t_f}\mathcal{L}\left(t,x,\dot{x},\dots,x^{(n-1)},x^{(n)}\right)dt$$

微分方程

$$\frac{\partial\mathcal{L}}{\partial x}-\frac{d}{dt}\left(\frac{\partial\mathcal{L}}{\partial\dot{x}}\right)+\frac{d^2}{dt^2}\left(\frac{\partial\mathcal{L}}{\partial\ddot{x}}\right)+\cdots+(-1)^n\frac{d^n}{dt^n}\left(\frac{\partial\mathcal{L}}{\partial\left(x^{(n)}\right)}\right)$$

■ 等式约束条件泛函极值

➢ Lagrange乘子与增广泛函

➢ 代数方程约束$f(x,t)=0$

➢ 微分方程约束$f(x,\dot{x},t)=0$

目标函数

$$\mathcal{J}=\int_{t_0}^{t_f}F(t,x,\dot{x})dt$$

微分方程

$$\frac{\partial F}{\partial x}+\left(\frac{\partial f}{\partial x}\right)^T\lambda-\frac{d}{dt}\left(\frac{\partial F}{\partial\dot{x}}\right)=0$$

■ (不)等式泛函极值与KKT方程

➢ 等式约束$f(x,t)=0$

➢ 不等式约束$g(x,t)\le 0$

目标函数

$$\mathcal{J}=\int_{t_0}^{t_f}F(t,x)dt$$

微分方程

$$\frac{\partial F}{\partial x}+\left(\frac{\partial f}{\partial x}\right)^T\lambda+\left(\frac{\partial g}{\partial x}\right)^T\mu=0$$

■ 一般等式泛函极值与Hamilton方程

■ 动态规划与Hamilton-Jacobi-Bellman方程

- **微分方程与函数空间**

  - 轨迹平滑：位置连续与速度连续
  - 轨迹平滑的目标函数：$\mathcal{L}\left(t, x, \dot{x}, \ldots, x^{(m-1)}, x^{(m)}\right)=\left(x^{(m)}\right)^2$
  - 微分方程构成的函数空间$x^{(n)}=0$
    - n阶函数空间：线性无关的函数集合$\{\varphi_i(t)\}_{i=0}^{n}$
    - 子空间：元素$\varphi(t)\in\Phi=span\{\varphi_0(t),\varphi_1(t),\ldots,\varphi_n(t)\}$
    - 线性表达：$\varphi(t)=a_0\varphi_0(t)+a_1\varphi_1(t)+\cdots+a_n\varphi_n(t)$
  - 函数逼近
    - 对于任何函数$f(t)$，在子空间$\Phi$中寻找一个元素$\varphi^*(t)\in\Phi$，使得$f(t)-\varphi^*(t)$在某个意义下最小
    - 正交多项式是函数逼近最常用的工具
    - 曲线拟合：多项式、三角函数等

$$x^{(2m)}=0$$

# 04 轨迹拟合 & 轨迹插值



- **函数优化与代数方程**
  - 梯形配置法
    - 梯形积分公式
    - 控制函数线性插值
    - 系统动力学线性插值
    - 状态(轨迹)函数二次插值

$$I_1(f) = \frac{b-a}{2}\left(f(a) + f(b)\right)$$

$$u(t) \approx u_k + \frac{\tau}{h_k}(u_{k+1} - u_k)$$

$$f(t) = \dot{x}(t) \approx f_k + \frac{\tau}{h_k}(f_{k+1} - f_k)$$

$$x(t) = \int \dot{x}(t)\, d\tau \approx c + f_k\tau + \frac{\tau^2}{2h_k}(f_{k+1} - f_k)$$

$$x(t) \approx x_k + f_k\tau + \frac{\tau^2}{2h_k}(f_{k+1} - f_k)$$

---

  - Simpson配置法
    - 梯形积分公式
    - 控制函数二次插值
    - 系统动力学二次插值
    - 状态(轨迹)函数三次插值

$$I_2(f) = \frac{b-a}{6}\left(f(a) + 4f\left(\frac{a+b}{2}\right) + f(b)\right)$$

$$u(t) = \frac{2}{h_k^2}\left(\tau - \frac{h_k}{2}\right)(\tau - h_k)u_k - \frac{4}{h_k^2}(\tau)(\tau - h_k)u_{k+\frac{1}{2}} + \frac{2}{h_k^2}(\tau)\left(\tau - \frac{h_k}{2}\right)u_{k+1}.$$

$$f(t) = \dot{x} = \frac{2}{h_k^2}\left(\tau - \frac{h_k}{2}\right)(\tau - h_k)f_k - \frac{4}{h_k^2}(\tau)(\tau - h_k)f_{k+\frac{1}{2}} + \frac{2}{h_k^2}(\tau)\left(\tau - \frac{h_k}{2}\right)f_{k+1}.$$

$$x(t) = x_k + f_k\left(\frac{\tau}{h_k}\right) + \frac{1}{2}\left(-3f_k + 4f_{k+\frac{1}{2}} - f_{k+1}\right)\left(\frac{\tau}{h_k}\right)^2 + \frac{1}{3}\left(2f_k - 4f_{k+\frac{1}{2}} + 2f_{k+1}\right)\left(\frac{\tau}{h_k}\right)^3.$$

- **代数方程与插值空间**

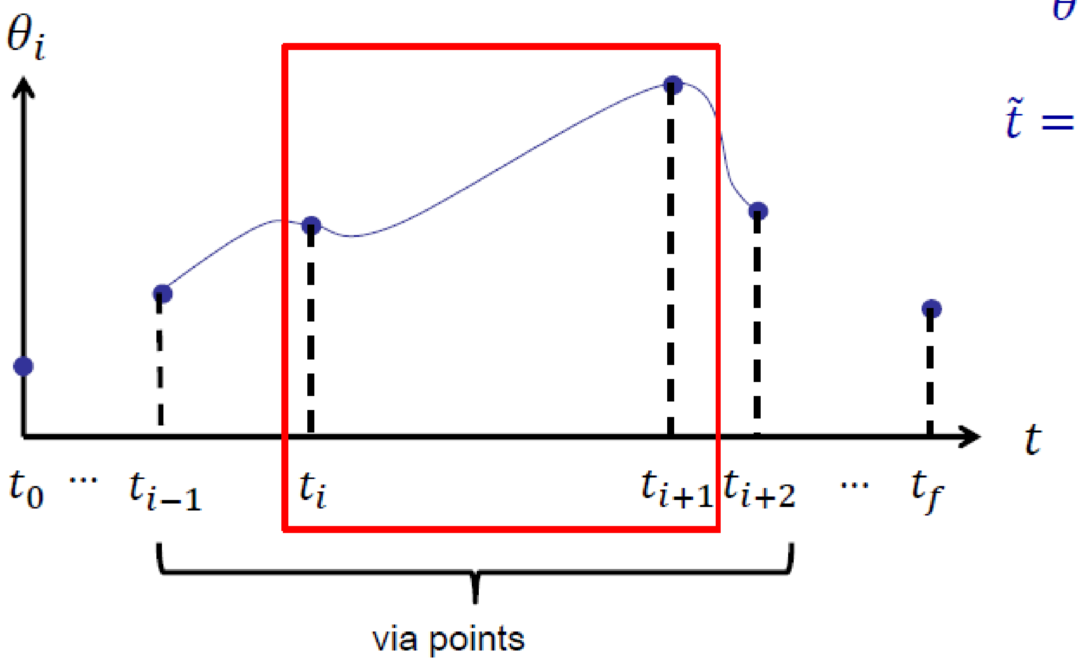
  - 轨迹光滑：插值点的光滑
  - 轨迹平滑的目标：满足插值点的光滑性
  - 代数方程构成的函数空间
    - n(+1)插值点：$t_0, t_1, \dots, t_n$
    - 子空间$\Phi$：不超过n次的代数多项式
  - 函数插值
    - 对于任何函数$f(t)$，在子空间$\Phi$中寻找一个元素$P(t) \in \Phi$，使得$P(t)$满足插值点约束
    - 曲线插值：三次插值、LFPB(Linear Function with Parabolic Blends)等

## ● 轨迹插值——三次插值

■ 轨迹插值：不同轨迹区间$[t_i, t_{i+1}]$以不同参数的函数来规划

■ 轨迹平滑：定义各函数的边界条件（包括位置和速度：$\theta(t_i)$，$\dot{\theta}(t_i)$，$\theta(t_{i+1})$，$\dot{\theta}(t_{i+1})$，有4个条件）

■ 轨迹算法：三次多项式




$$\theta(\tilde{t}) = a_0 + a_1\tilde{t} + a_2\tilde{t}^2 + a_3\tilde{t}^3 \qquad t \in [t_i, t_{i+1}]$$

$$\tilde{t} = t - t_i \quad so\ \tilde{t}|_{t=t_i} = 0 \ \text{ and } \ \tilde{t}|_{t=t_{i+1}} \equiv \Delta t = t_{i+1} - t_i > 0$$

$$\theta(\tilde{t}|_{t=t_i}) = \theta_i = a_0$$

$$\theta(\tilde{t}|_{t=t_{i+1}}) = \theta_{i+1} = a_0 + a_1\Delta t + a_2\Delta t^2 + a_3\Delta t^3$$

$$\dot{\theta}\left(\tilde{t}|_{t=t_i}\right) = \dot{\theta}_i = a_1$$

$$\dot{\theta}\left(\tilde{t}|_{t=t_{i+1}}\right) = \dot{\theta}_{i+1} = a_1 + 2a_2\Delta t + 3a_3\Delta t^2$$

- **轨迹插值——三次样条**

■ 轨迹插值：不同轨迹区间$[t_i, t_{i+1}]$以不同参数的三次多项式函数来规划

■ 轨迹平滑：加速度连续

➢ $2N$个边界条件：$\theta(t_0)$与$\theta(t_1)$，$\theta(t_1)$与$\theta(t_2)$，…，$\theta(t_{N-1})$与$\theta(t_N)$

➢ $2(N-1)$个速度和加速度连续条件$\dot{\theta}(t_i) = \dot{\theta}(t_{i+1})$，$\ddot{\theta}(t_i) = \ddot{\theta}(t_{i+1})$

➢ 2个附加条件：

➢ (1) Natural三次样条：$\ddot{\theta}(t_0) = \ddot{\theta}(t_N) = 0$

➢ (2) Clamp三次样条：$\dot{\theta}(t_0) = u, \dot{\theta}(t_N) = v$

➢ (3)运动的周期性（用得不多）

■ 轨迹算法：三次样条，$4N$个系数，$4N$个约束条件(方程)，唯一解

林沛群. 机器人学之运动学，国立台湾大学.

- **轨迹拟合 & 轨迹插值**

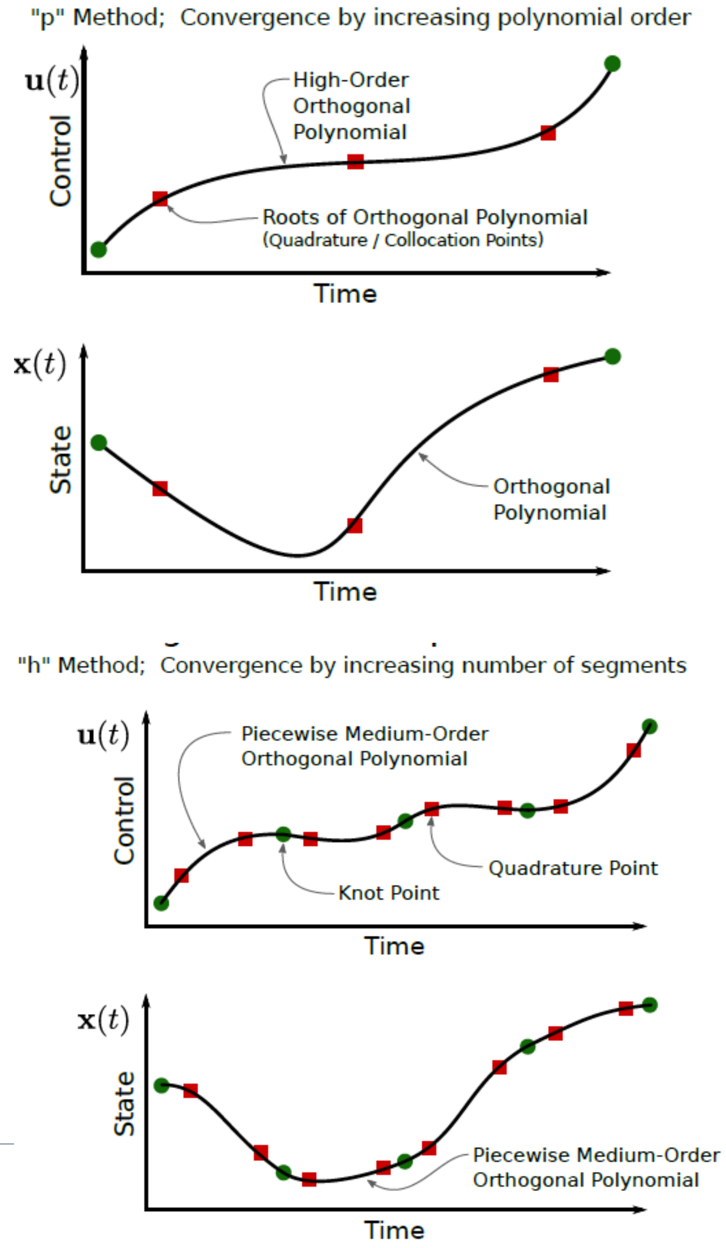
| 轨迹拟合算法 | 形式 | 微分方程 | 约束个数 |
|---|---|---|---|
| 最小加速度 | $x(t)=a_0+a_1t+a_2t^2+a_3t^3$ | $x^{(4)}=0$ | 4 |
| 最小jerk | $x(t)=a_0+a_1t+a_2t^2+a_3t^3+a_4t^4+a_5t^5$ | $x^{(6)}=0$ | 6 |
| 最小snap | $x(t)=a_0+a_1t+a_2t^2+a_3t^3+a_4t^4+a_5t^5+a_6t^6+a_7t^7$ | $x^{(8)}=0$ | 8 |
| 正弦轨迹 | $x(t)=a_0+a_1\cos a_2t+a_3\sin a_2t$ | $x^{(4)}=0$ | 4 |
| 摆线轨迹 | $x(t)=a_0+a_1t-a_2\sin a_3t$ | $x^{(4)}=0$ | 4 |
| **轨迹插值算法** | **形式** | **代数方程** | **约束个数** |
| LFPB | Linear & Parabolic结合 | | 3N |
| 三次样条 | $x_j(t)=a_{j0}+a_{j1}t+a_{j2}t^2+a_{j3}t^3$ | | 4N |

# 04 轨迹拟合 & 轨迹插值



- **p方法和h方法**

  - 解析解
    - 如何确定解析解(轨迹拟合)的收敛性?
    - $x(t) = a_0 + a_1 t + a_2 t^2 + \cdots + a_n t^n$
    - p方法：增加阶次n
    - 存在的问题：Runge现象(过拟合，加速度过大)
  - 数值解
    - 分段多项式
    - 使用多段低阶次多项式代替单段高阶次多项式
    - h方法：增加分段次数N




https://arxiv.org/pdf/1707.00284.pdf

- **轨迹优化——求解方法汇总**


轨迹优化求解方法
解析法
无约束泛函极值Euler-Lagrange方程
等式约束泛函极值Lagrange乘子
不等式约束泛函极值KKT方程
动态规划HJB方程
...
微分方程
轨迹拟合
数值法
梯形配置法二次插值
Simpson配置法三次插值
...
代数方程
轨迹插值