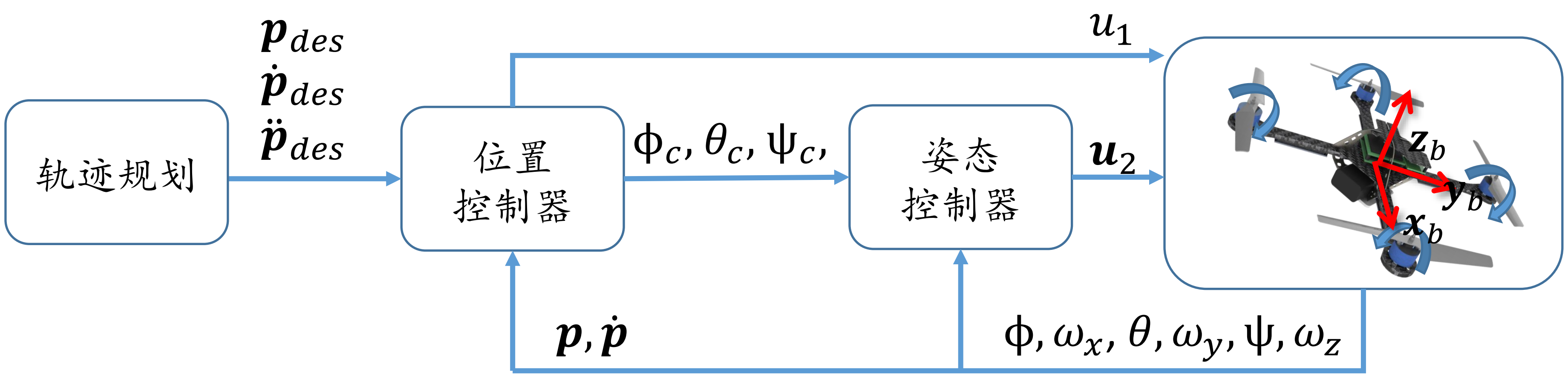
线性控制器



- 位置控制

PID: $$\ddot{\boldsymbol{p}}_{i,c} = \ddot{\boldsymbol{p}}_i^{\,des} + K_{d,i}\left(\dot{\boldsymbol{p}}_i^{\,des} - \dot{\boldsymbol{p}}_i\right) + K_{p,i}\left(\boldsymbol{p}_i^{\,des} - \boldsymbol{p}_i\right)$$

模型: $$u_1 = m(g + \ddot{\boldsymbol{p}}_{3,c})$$

(牛顿方程)

$$\phi_c = \frac{1}{g}(\ddot{\boldsymbol{p}}_{1,c} sin\boxed{\psi} - \ddot{\boldsymbol{p}}_{2,c} \cos\boxed{\psi}) \qquad \theta_c = \frac{1}{g}(\ddot{\boldsymbol{p}}_{1,c} cos\boxed{\psi} + \ddot{\boldsymbol{p}}_{2,c} sin\boxed{\psi})$$

注意：这些是当前测量的yaw，不是期望的yaw

- 姿态控制

PID: $$\begin{bmatrix} \ddot{\phi}_c \\ \ddot{\theta}_c \\ \ddot{\psi}_c \end{bmatrix} = \begin{bmatrix} K_{p,\phi}(\phi_c - \phi) + K_{d,\phi}(\dot{\phi}_c - \dot{\phi}) \\ K_{p,\theta}(\theta_c - \theta) + K_{d,\phi}(\dot{\theta}_c - \dot{\theta}) \\ K_{p,\psi}(\psi_c - \psi) + K_{d,\psi}(\dot{\psi}_c - \dot{\psi}) \end{bmatrix}$$

模型: $$\boldsymbol{u}_2 = \boldsymbol{I} \cdot \begin{bmatrix} \ddot{\phi}_c \\ \ddot{\theta}_c \\ \ddot{\psi}_c \end{bmatrix} + \begin{bmatrix} \omega_x \\ \omega_y \\ \omega_z \end{bmatrix} \times \boldsymbol{I} \cdot \begin{bmatrix} \omega_x \\ \omega_y \\ \omega_z \end{bmatrix}$$ (欧拉方程)

总结

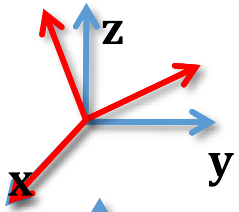
　两者都是串级控制，外环控制位置，内环控制姿态。但相比于Linear Controller，SE(3) Controller有两个特点：

1. 根据误差计算期望的推力向量，用其根据微分平坦计算姿态误差（用旋转矩阵表示而非欧拉角，且没有小角度假设）。
2. 推力控制输入是期望的推力向量投影到当前测量的姿态Z轴上的分量。

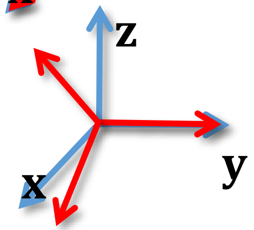
四旋翼动力学模型

欧拉角

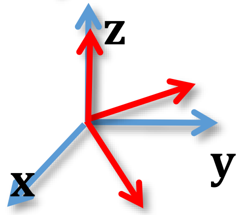
- $\boldsymbol{R}_x(\phi)=\begin{bmatrix}1 & 0 & 0\\ 0 & \cos\phi & -\sin\phi\\ 0 & \sin\phi & \cos\phi\end{bmatrix}$



- $\boldsymbol{R}_y(\theta)=\begin{bmatrix}\cos\theta & 0 & \sin\theta\\ 0 & 1 & 0\\ -\sin\theta & 0 & \cos\theta\end{bmatrix}$



- $\boldsymbol{R}_z(\psi)=\begin{bmatrix}\cos\psi & -\sin\psi & 0\\ \sin\psi & \cos\psi & 0\\ 0 & 0 & 1\end{bmatrix}$



$\boldsymbol{R}_z(\psi)\cdot\boldsymbol{R}_x(\phi)\cdot\boldsymbol{R}_y(\theta)\Rightarrow\boldsymbol{R}$

R: body to world (ZXY)
($\boldsymbol{\phi},\boldsymbol{\theta},\boldsymbol{\psi}$ 分别对应于X-Y-Z Euler)

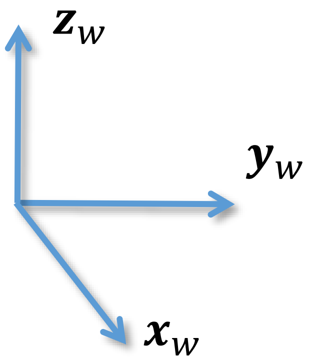
系统状态

$\boldsymbol{x}=[x\ \dot{x}\ y\ \dot{y}\ z\ \dot{z}\ \phi\ \dot{\phi}\ \theta\ \dot{\theta}\ \psi\ \dot{\psi}]$

系统输入

$u=[U_1U_2U_3U_4]$

合推力 X, Y, Z三轴力矩

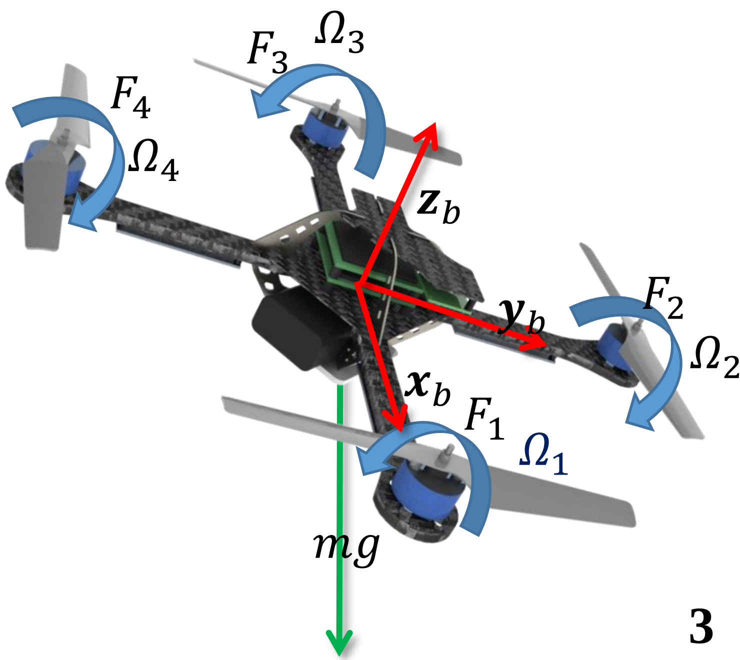


l是四轴飞行器中心与螺旋桨中心之间的距离，b和d分别是推力和反扭矩系数

$$\boldsymbol{u}=\begin{bmatrix}b & b & b & b\\ 0 & bl & 0 & -bl\\ -bl & 0 & bl & 0\\ d & -d & d & -d\end{bmatrix}\begin{bmatrix}\Omega_1^2\\ \Omega_2^2\\ \Omega_3^2\\ \Omega_4^2\end{bmatrix}$$

混控矩阵

螺旋桨的速度$\Omega_1,\Omega_2,\Omega_3,\Omega_4$



四旋翼动力学模型

由空气动力学可知四轴转桨推力F_1,F_2,F_3,F_4正比于螺旋桨的速度$\Omega_1,\Omega_2,\Omega_3,\Omega_4$的平方。
l是四轴飞行器中心与螺旋桨中心之间的距离，b和d分别是推力和反扭矩系数

$F_i = b{\Omega_i}^2$ 单桨推力

$U_1 = F_1 + F_2 + F_3 + F_4$ 合推力

$U_2 U_3 U_4$三轴力矩

J_r总转动惯量

I_x，I_y，I_z三轴转动惯量

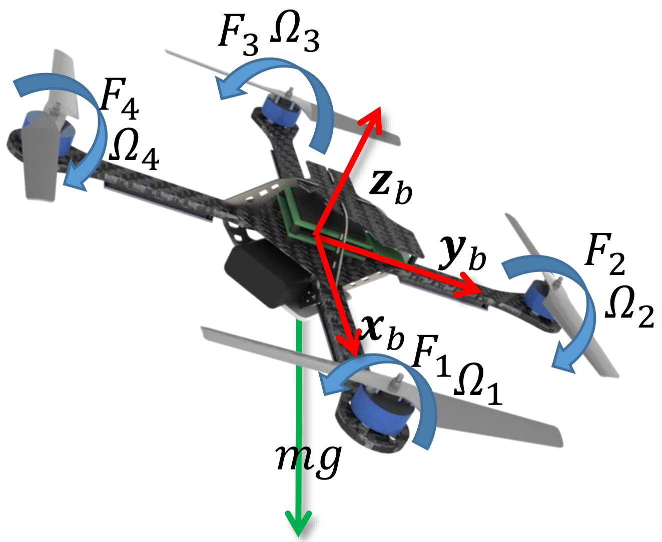
$$\boldsymbol{u} = \begin{bmatrix} b & b & b & b \\ 0 & bl & 0 & -bl \\ -bl & 0 & bl & 0 \\ d & -d & d & -d \end{bmatrix} \begin{bmatrix} \Omega_1^2 \\ \Omega_2^2 \\ \Omega_3^2 \\ \Omega_4^2 \end{bmatrix}$$

系统状态

$$\boldsymbol{x} = [x\ \dot{x}\ y\ \dot{y}\ z\ \dot{z}\ \phi\ \dot{\phi}\ \theta\ \dot{\theta}\ \psi\ \dot{\psi}]$$

系统输入

$$u = [U_1 U_2 U_3 U_4]$$



完整模型

$$\Omega_r = -\Omega_1 + \Omega_2 - \Omega_3 + \Omega_4$$

$$\ddot{x} = (sin\psi sin\phi + cos\phi sin\theta cos\psi)\frac{U_1}{m}$$

$$\ddot{y} = (-cos\psi sin\phi + sin\psi sin\theta cos\phi)\frac{U_1}{m}$$

$$\ddot{z} = -g + (cos\theta cos\phi)\frac{U_1}{m}$$

$$\ddot{\phi} = \frac{I_y - I_z}{I_x}\dot{\theta}\dot{\psi} - \frac{J_r}{I_x}\dot{\theta}\,\Omega_r + \frac{U_2}{I_x}$$

$$\ddot{\theta} = \frac{I_z - I_x}{I_y}\dot{\phi}\dot{\psi} + \frac{J_r}{I_y}\dot{\phi}\Omega_r + \frac{U_3}{I_y}$$

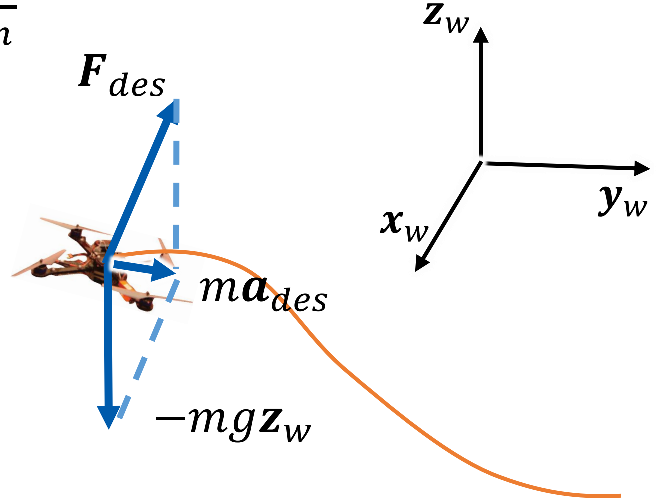
$$\ddot{\psi} = \frac{I_x - I_y}{I_z}\dot{\phi}\dot{\theta} + \frac{U_4}{I_z}$$

- **线性简化模型**　平衡悬停态　$(\phi_0 \sim 0, \theta_0 \sim 0, u_{1,0} \sim mg)$

牛顿方程　$m\ddot{\boldsymbol{p}} = \begin{bmatrix} 0 \\ 0 \\ -mg \end{bmatrix} + \boldsymbol{R} \begin{bmatrix} 0 \\ 0 \\ F_1 + F_2 + F_3 + F_4 \end{bmatrix}$ $\Rightarrow$

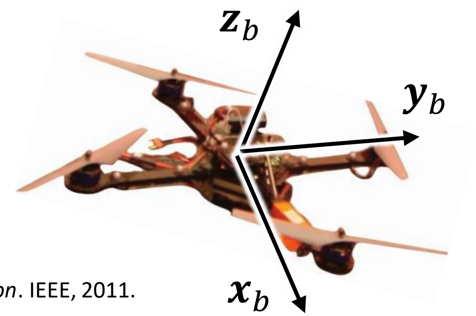
$$\ddot{\boldsymbol{p}}_1 = \ddot{x} = g(\theta cos\psi + \phi sin\psi)$$
$$\ddot{\boldsymbol{p}}_2 = \ddot{y} = g(\theta sin\psi - \phi cos\psi)$$
$$\ddot{\boldsymbol{p}}_3 = \ddot{z} = -g + \frac{u_1}{m}$$

$$\boldsymbol{R} = \begin{bmatrix} c\psi c\theta - s\phi s\psi s\theta & -c\phi s\psi & c\psi s\theta + c\theta s\phi s\psi \\ c\theta s\psi + c\psi s\phi s\theta & c\phi c\psi & s\psi s\theta - c\psi c\theta s\phi \\ -c\phi s\theta & s\phi & c\phi c\theta \end{bmatrix}$$



欧拉角微分　$\begin{bmatrix} \omega_x \\ \omega_y \\ \omega_z \end{bmatrix} = \begin{bmatrix} c\theta & 0 & -c\phi s\theta \\ 0 & 1 & s\phi \\ s\theta & 0 & c\phi c\theta \end{bmatrix} \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix}$ $\Rightarrow$ $\begin{bmatrix} \omega_x \\ \omega_y \\ \omega_z \end{bmatrix} = \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix}$

欧拉方程：　$\boldsymbol{I} \cdot \begin{bmatrix} \ddot{\phi} \\ \ddot{\theta} \\ \ddot{\psi} \end{bmatrix} + \begin{bmatrix} \omega_x \\ \omega_y \\ \omega_z \end{bmatrix} \times \boldsymbol{I} \cdot \begin{bmatrix} \omega_x \\ \omega_y \\ \omega_z \end{bmatrix} = \begin{bmatrix} l(F_2 - F_4) \\ l(F_3 - F_1) \\ M_1 - M_2 + M_3 - M_4 \end{bmatrix} = \begin{bmatrix} U_2 \\ U_3 \\ U_4 \end{bmatrix}$



Mellinger, Daniel, and Vijay Kumar. "Minimum snap trajectory generation and control for quadrotors." *2011 IEEE international conference on robotics and automation*. IEEE, 2011.

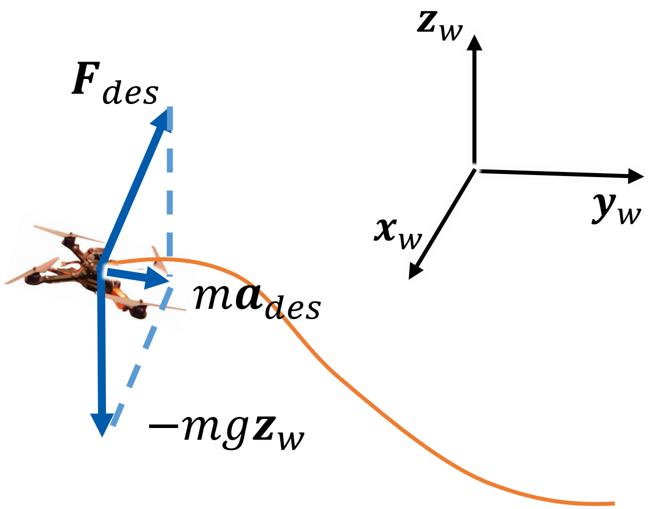
反应式控制

控制目标:

- 理想情况：

$$m\boldsymbol{a} = \boldsymbol{F} - mg\boldsymbol{z}_w$$

$$\boldsymbol{F}_{des} = mg\boldsymbol{z}_w + m\boldsymbol{a}_{des} \qquad \boldsymbol{z}_{b,des} = \frac{\boldsymbol{F}_{des}}{||\boldsymbol{F}_{des}||}$$



- 现实情况：

$$\boldsymbol{q}_{des} = \boldsymbol{q}_z \otimes \boldsymbol{q}_\psi$$

期望推力与姿态 → 自驾仪 Autopilot

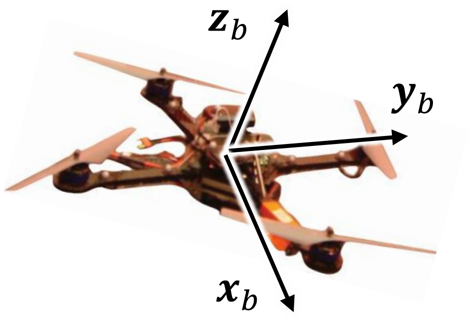
$$\boldsymbol{e}_p = \boldsymbol{p} - \boldsymbol{p}_{des},\ \boldsymbol{e}_v = \boldsymbol{v} - \boldsymbol{v}_{des}$$

位置与速度误差

$$\boldsymbol{F}_{des} = -K_p\boldsymbol{e}_p - K_v\boldsymbol{e}_v + mg\boldsymbol{z}_w + m\boldsymbol{a}_{des}$$

增益

反应式控制



Mellinger, Daniel, and Vijay Kumar. "Minimum snap trajectory generation and control for quadrotors." *2011 IEEE international conference on robotics and automation*. IEEE, 2011.

反应式控制

优势

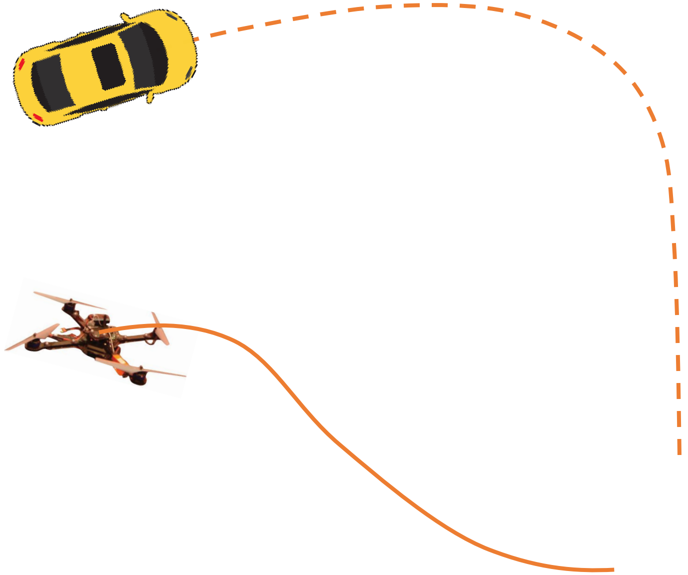
- 易于实现
- 考虑误差

缺陷

- 对于复杂系统实现Non-trivial
- 增益项手动调整
- 对于耦合系统与约束无处理
- 忽略未来的决定

腿式机器人:
太过复杂

无人机，车:
分解为姿态位置或
横向纵向控制器，
忽略状态耦合



Mellinger, Daniel, and Vijay Kumar. "Minimum snap trajectory generation and control for quadrotors." *2011 IEEE international conference on robotics and automation*. IEEE, 2011.

反应式控制

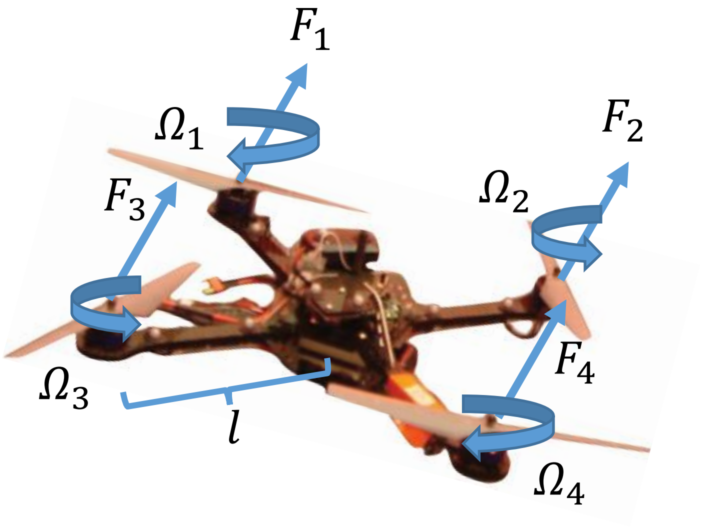
- 最优控制
  - 系统模型

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

$$\boldsymbol{x}_{k+1} = f_d(\boldsymbol{x}_k, \boldsymbol{u}_k) \qquad \boldsymbol{x}_0 \text{ 初始条件}$$

状态　　输入

  - 动力学模型



U_1 合推力

$U_2U_3U_4$三轴力矩

系统状态: $\begin{bmatrix} x \\ \dot{x} \\ y \\ \dot{y} \\ z \\ \dot{z} \\ \phi \\ \dot{\phi} \\ \theta \\ \dot{\theta} \\ \psi \\ \dot{\psi} \end{bmatrix}$　系统输入: $\begin{bmatrix} U_1 \\ U_2 \\ U_3 \\ U_4 \end{bmatrix}$

$$\ddot{x} = (sin\psi sin\phi + cos\phi sin\theta cos\psi)\frac{U_1}{m}$$

$$\ddot{y} = (-cos\psi sin\phi + sin\psi sin\theta cos\phi)\frac{U_1}{m}$$

$$\ddot{z} = -g + (cos\theta cos\phi)\frac{U_1}{m}$$

$$\ddot{\phi} = \frac{I_y - I_z}{I_x}\dot{\theta}\dot{\psi} - \frac{J_r}{I_x}\dot{\theta}\Omega_r + \frac{U_2}{I_x}$$

$$\ddot{\theta} = \frac{I_z - I_x}{I_y}\dot{\phi}\dot{\psi} + \frac{J_r}{I_y}\dot{\phi}\Omega_r + \frac{U_3}{I_y}$$

$$\ddot{\psi} = \frac{I_x - I_y}{I_z}\dot{\phi}\dot{\theta} + \frac{U_4}{I_z}$$

$$\dot{\boldsymbol{x}} = f_c(\boldsymbol{x}, u) = \begin{pmatrix} \dot{x} \\ u_x \frac{1}{m} U_1 \\ \dot{y} \\ u_y \frac{1}{m} U_1 \\ \dot{z} \\ -g + (\cos\theta\cos\phi)\frac{1}{m}U_1 \\ \dot{\phi} \\ a_1\dot{\theta}\dot{\psi} - a_2\dot{\theta}\Omega_r + b_1 U_2 \\ \dot{\theta} \\ a_3\dot{\phi}\dot{\psi} + a_4\dot{\phi}\Omega_r + b_2 U_3 \\ \dot{\psi} \\ a_5\dot{\phi}\dot{\theta} + b_3 U_4 \end{pmatrix}$$

$$a_1 = \frac{I_y - I_z}{I_x} \quad a_2 = \frac{J_r}{I_x} \quad a_3 = \frac{I_z - I_x}{I_y} \quad a_4 = \frac{J_r}{I_y} \quad a_5 = \frac{I_x - I_y}{I_z}$$

$$b_1 = \frac{1}{I_x} \quad b_2 = \frac{1}{I_y} \quad b_3 = \frac{1}{I_z}$$

$$u_x = (\sin\psi\sin\phi + \cos\psi\sin\theta\cos\phi)$$

$$u_y = (-\cos\psi\sin\phi + \sin\psi\sin\theta\cos\phi)$$

最优控制

- 目标最小化函数 $$\min_{u_{0:N-1}} \underbrace{\sum_{k=0}^{N-1} q(\boldsymbol{x}_k, \boldsymbol{u}_k)}_{\text{阶段代价}} + \underbrace{p(\boldsymbol{x}_N)}_{\text{终端代价}}$$

阶段代价（stage cost）　终端代价（terminal cost）

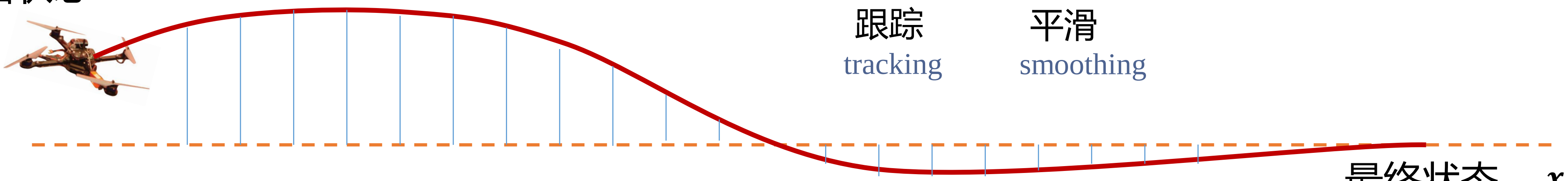
- 跟踪任务(tracking)举例：

$\boldsymbol{e}_k = \boldsymbol{x}_k - \boldsymbol{x}_k^{ref}$

$q(\boldsymbol{x}_k, \boldsymbol{u}_k) = ||\boldsymbol{e}_k||^2 + \rho \Delta \boldsymbol{u}_k^2$

跟踪 tracking　平滑 smoothing

初始状态 $\boldsymbol{x}_0$



最终状态 $\boldsymbol{x}_N$

$p(\boldsymbol{x}_N) = w_N ||\boldsymbol{e}_N||^2$

最优控制

- 目标最小化函数

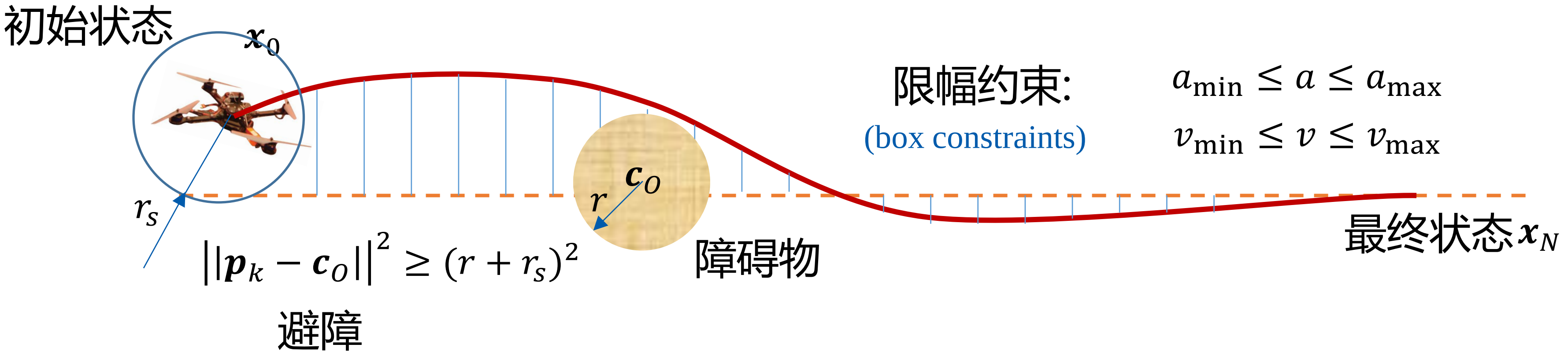
$$\min_{u_{0:N-1}} \sum_{k=0}^{N-1} \underbrace{q(\boldsymbol{x}_k, \boldsymbol{u}_k)}_{\text{阶段代价 (stage cost)}} + \underbrace{p(\boldsymbol{x}_N)}_{\text{终端代价 (terminal cost)}}$$

- 约束项

$$\boldsymbol{x}_{k+1} = f_d(\boldsymbol{x}_k, \boldsymbol{u}_k)$$

$h(\boldsymbol{x}_k, \boldsymbol{u}_k) = \boldsymbol{0}$ 等式约束

$g(\boldsymbol{x}_k, \boldsymbol{u}_k) \leq \boldsymbol{0}$ 不等式约束



限幅约束:
(box constraints)

$a_{\min} \leq a \leq a_{\max}$

$v_{\min} \leq v \leq v_{\max}$

- 最优控制
  - 目标最小化函数

$$\min_{u_{0:N-1}} \underbrace{\sum_{k=0}^{N-1} q(\boldsymbol{x}_k, \boldsymbol{u}_k)}_{\substack{\text{阶段代价} \\ \text{(stage cost)}}} + \underbrace{p(\boldsymbol{x}_N)}_{\substack{\text{终端代价} \\ \text{(terminal cost)}}}$$

  - 约束项

$$\boldsymbol{x}_{k+1} = f_d(\boldsymbol{x}_k, \boldsymbol{u}_k)$$

$h(\boldsymbol{x}_k, \boldsymbol{u}_k) = \boldsymbol{0}$ 等式约束

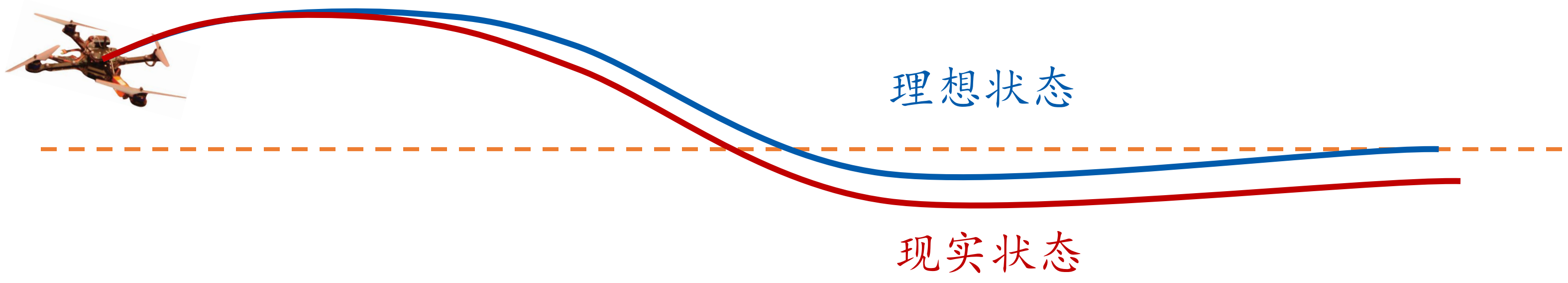
$g(\boldsymbol{x}_k, \boldsymbol{u}_k) \leq \boldsymbol{0}$ 不等式约束

  - 最优解

$\boldsymbol{z}^* = [\boldsymbol{u}_0{}^T, \dots, \boldsymbol{u}_{N-1}{}^T]^T$ 最优序列

理想情况下用最优解$\boldsymbol{z}^*$作为系统的控制输入!

- 开环最优控制难点
  - 系统模型不准确，长时间下累计更多误差
  - 最优解z^*无法被准确执行
  - 较长的预测时域使得问题难以求解
  - 系统可能受到外界扰动



模型预测控制概念

- 使用动力学模型预测未来走向选择最佳的控制输入

有限的预测时域

- 测量信息的反馈

- 测量信息的反馈

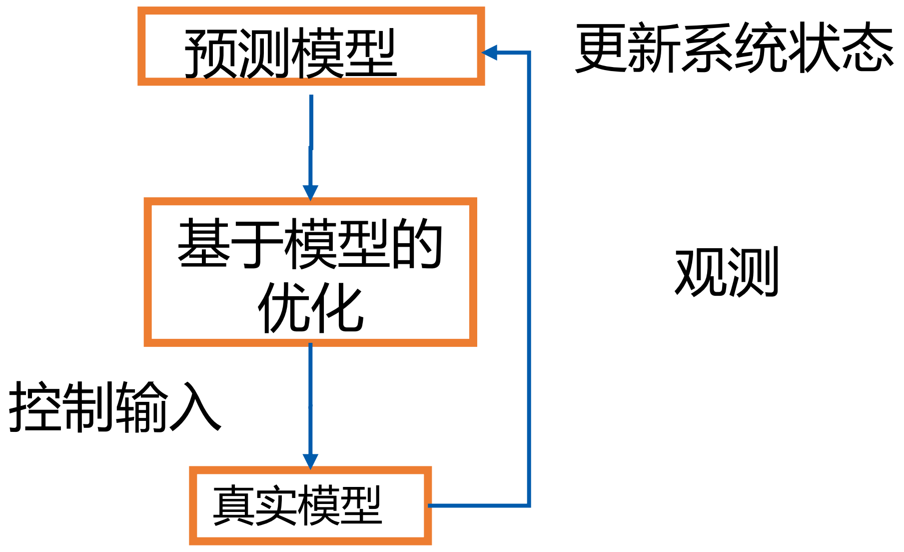
→从估计的当前状态开始

- 优化最佳的控制序列

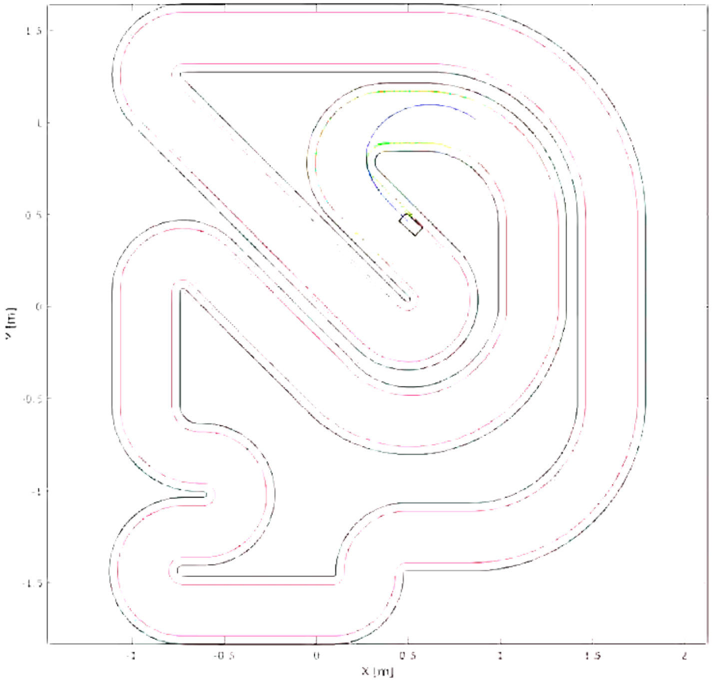
→在有限的预测时域内找到最佳的控制输入序列

- 滚动预测框架

→执行第一个最佳的控制输入，更新状态重规划



- 滚动优化框架



Liniger, Alexander, Alexander Domahidi, and Manfred Morari. "Optimization-based autonomous racing of 1: 43 scale RC cars." Optimal Control Applications and Methods 36.5 (2015): 628-647.

模型预测控制概念

- 使用动力学模型预测未来走向选择最佳的控制输入
- 跟踪目标为例

$$\min_{u_0,u_1,\ldots,u_N} \sum_{k}^{N} ||\mathrm{p}_k - \boldsymbol{r}(t)||^2 + \rho\Delta\boldsymbol{u}_k^2$$

s. t.

$$\boldsymbol{x}_{k+1} = f(\boldsymbol{x}_k, \boldsymbol{u}_k)$$
$$\boldsymbol{y}_k = g(\boldsymbol{x}_k)$$

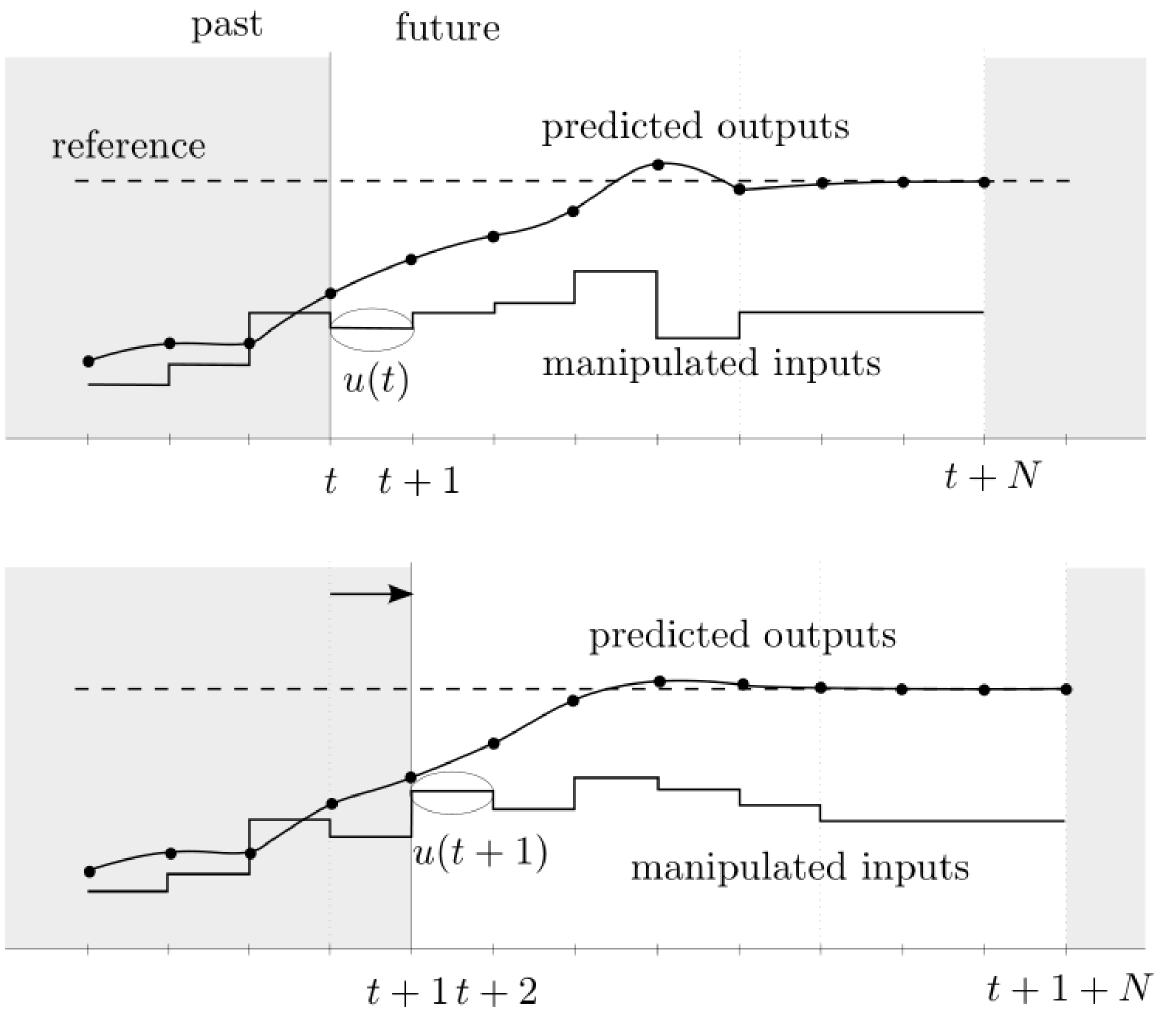
预测模型

$$\boldsymbol{x}_{min} \le \boldsymbol{x}_k \le \boldsymbol{x}_{max}$$
$$\boldsymbol{u}_{min} \le \boldsymbol{u}_k \le \boldsymbol{u}_{max}$$

约束

$$\boldsymbol{x}_0 = \boldsymbol{x}(t)$$

状态反馈



Borrelli, Francesco, Alberto Bemporad, and Manfred Morari. Predictive control for linear and hybrid systems. Cambridge University Press, 2017.

模型预测控制概念

- 别名
  - 开环最优反馈(Open Loop Optimal Feedback)
  - 反应式规划(Reactive Scheduling)
  - 滚动优化控制(Receding Horizon Control)
- 优势
  - 考虑未来时域（尽管有限）
  - 考虑误差
  - 减小问题规模（求解器一般有warm-start，由上次最优解开始作为初值）

模型预测控制案例

- **机器人当中运用**

非线性MPC用于容错控制

Whole-body MPC 用于轮腿式机器人

Nan, Fang, et al. "Nonlinear MPC for Quadrotor Fault-Tolerant Control." arXiv preprint arXiv:2109.12886 (2021).
Bjelonic, Marko, et al. "Whole-body mpc and online gait sequence generation for wheeled-legged robots." 2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). IEEE, 2020.

模型预测控制参数

- MPC的参数选择
- 模型的选择。效率与准确性之间权衡

$$\dot{\boldsymbol{x}} = f_c(\boldsymbol{x}, u) = \begin{pmatrix} \dot{z} \\ -g + \frac{1}{m}U_1 \cos\phi \\ \dot{y} \\ -\frac{1}{m}U_1 \sin\phi \\ \dot{\phi} \\ \frac{1}{I_x}U_2 \end{pmatrix}$$

VS

$$\dot{\boldsymbol{x}} = f_c(\boldsymbol{x}, u) = \begin{pmatrix} \dot{x} \\ u_x \frac{1}{m}U_1 \\ \dot{y} \\ u_y \frac{1}{m}U_1 \\ \dot{z} \\ -g + (\cos\theta\cos\phi)\frac{1}{m}U_1 \\ \dot{\phi} \\ a_1\dot{\theta}\dot{\psi} - a_2\dot{\theta}\Omega_r + b_1U_2 \\ \dot{\theta} \\ a_3\dot{\phi}\dot{\psi} + a_4\dot{\phi}\Omega_r + b_2U_3 \\ \dot{\psi} \\ a_5\dot{\phi}\dot{\theta} + b_3U_4 \end{pmatrix}$$

平衡悬停态 $\theta = \dot{\theta} = \psi = \dot{\psi} \sim 0$

系统状态$\boldsymbol{x} = [z, \dot{z}, y, \dot{y}\ \phi, \dot{\phi}\ ]$
系统输入$\boldsymbol{u} = [U_1, U_2\ ]$

系统状态$\boldsymbol{x} = [x, \dot{x}, y, \dot{y}, z, \dot{z}, \phi, \dot{\phi}, \theta,\ \dot{\theta}\ , \psi, \dot{\psi}]$

系统输入$\boldsymbol{u} = [U_1, U_2, U_3, U_4]$

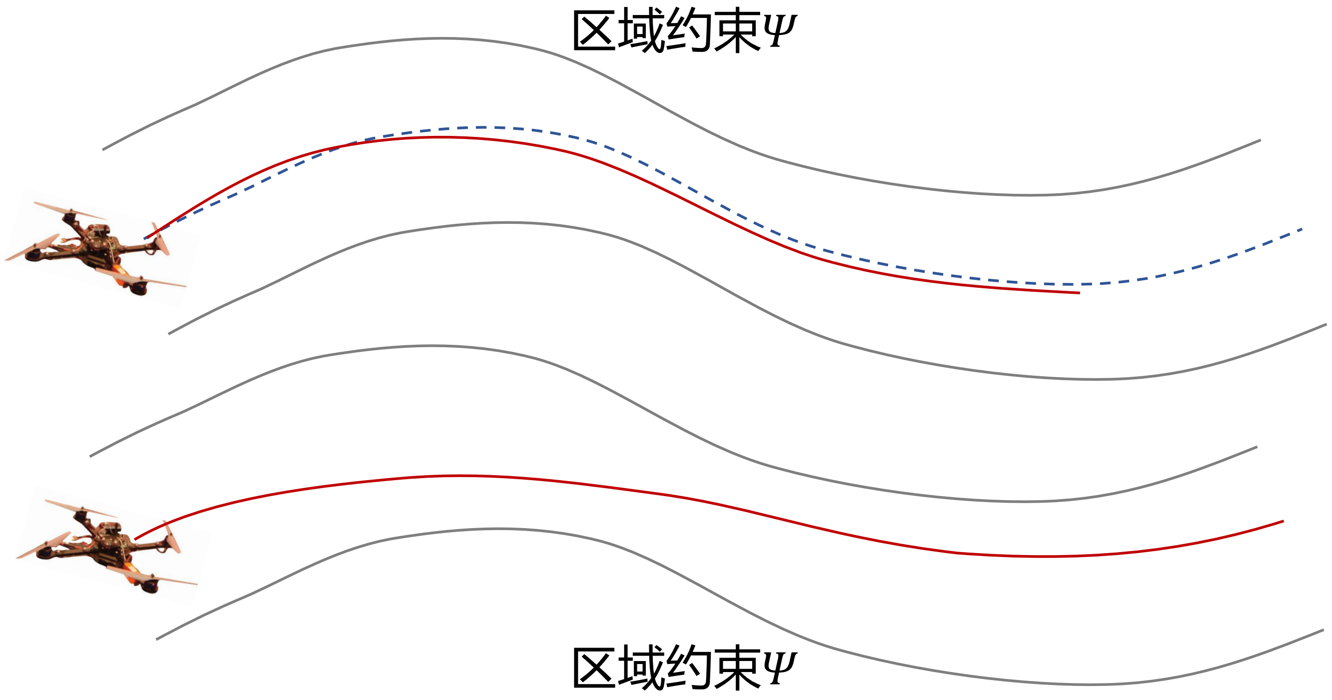
二维简化模型
简易性

完整状态模型
准确性

模型预测控制参数

- MPC的参数选择
- 代价函数的选择

控制器与规划器



$$\min_{u_0,u_1,\ldots,u_N} \sum_{k}^{N} ||\mathrm{p}_k - \boldsymbol{r}(t)||^2 + \rho\Delta\boldsymbol{u}_k^2$$

最小化跟踪误差（控制器）

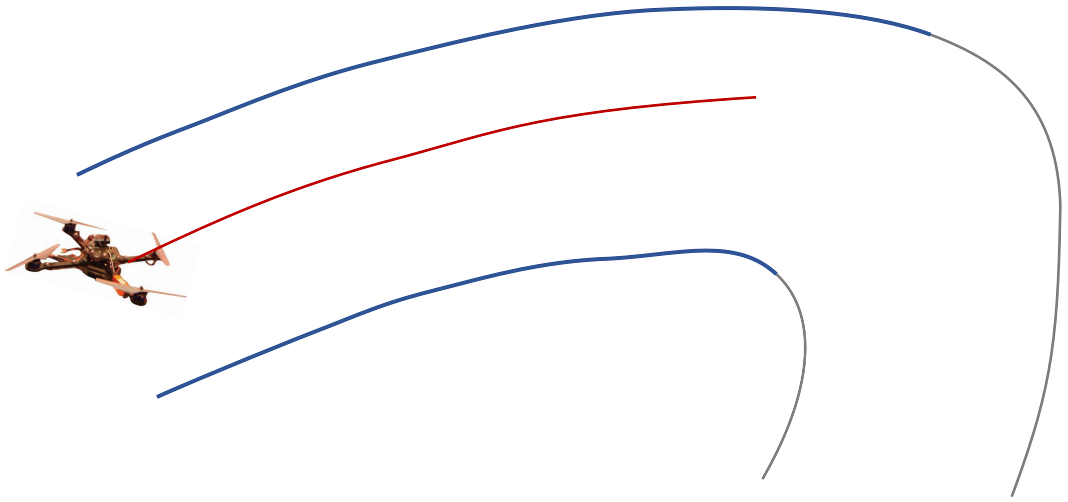
$$\min_{u(t)} \sum_{k}^{N} \rho\Delta\boldsymbol{u}_k^2 + \gamma T$$

$$\text{st. } \mathrm{p}(t) \in \Psi$$

时间参数化控制指令
最小化时间（规划器）

模型预测控制参数

- MPC的参数选择
- 预测时域：计算量与解的可行性的权衡



- 较短的预测时域
  - 较小的计算量
  - 次优的解（可能不安全）
- 较长的预测时域
  - 较大的计算量
  - 更优更安全的解

模型预测控制参数

预测时长（Prediction horizon）的长短?

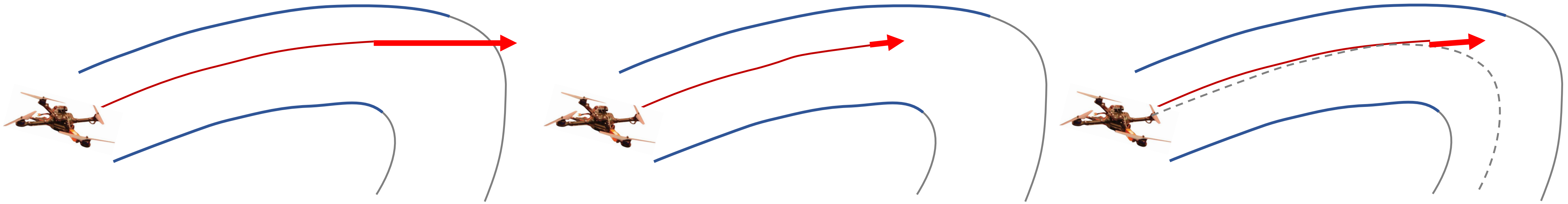
→ 权衡 计算时间computation overload 和递归可行性 recursive feasibility

- 较短的预测时长
  - 减少计算时间
  - 短视性（不安全）
- 末状态增加某些额外的约束可以提高recursive feasibility.

末速度无额外约束

末速度限制为0

末速度限制为全局轨迹该位置附近的速度



- 线性时变模型Linear Time-Varying (LTV) model

$$\boldsymbol{x}_{k+1} = \boldsymbol{A}\boldsymbol{x}_k + \boldsymbol{B}\boldsymbol{u}_k \rightarrow \boldsymbol{x}_{k+1} = \underline{\boldsymbol{A}_k(t)}\boldsymbol{x}_k + \underline{\boldsymbol{B}_k(t)}\boldsymbol{u}_k$$

- 非线性模型

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

- 通过线性化非线性模型得到LTV

$$\dot{\boldsymbol{x}} \approx f(\overline{\boldsymbol{x}}, \overline{\boldsymbol{u}}) + \left.\frac{\partial f}{\partial \boldsymbol{x}}\right|_{\overline{\boldsymbol{x}}, \overline{\boldsymbol{u}}} (\boldsymbol{x} - \overline{\boldsymbol{x}}) + \left.\frac{\partial f}{\partial \boldsymbol{u}}\right|_{\overline{\boldsymbol{x}}, \overline{\boldsymbol{u}}} (\boldsymbol{u} - \overline{\boldsymbol{u}})$$

$$\dot{\boldsymbol{x}} = \boldsymbol{A}_c\boldsymbol{x} + \boldsymbol{B}_c\boldsymbol{u} + \boldsymbol{g}_c$$

离散化求解

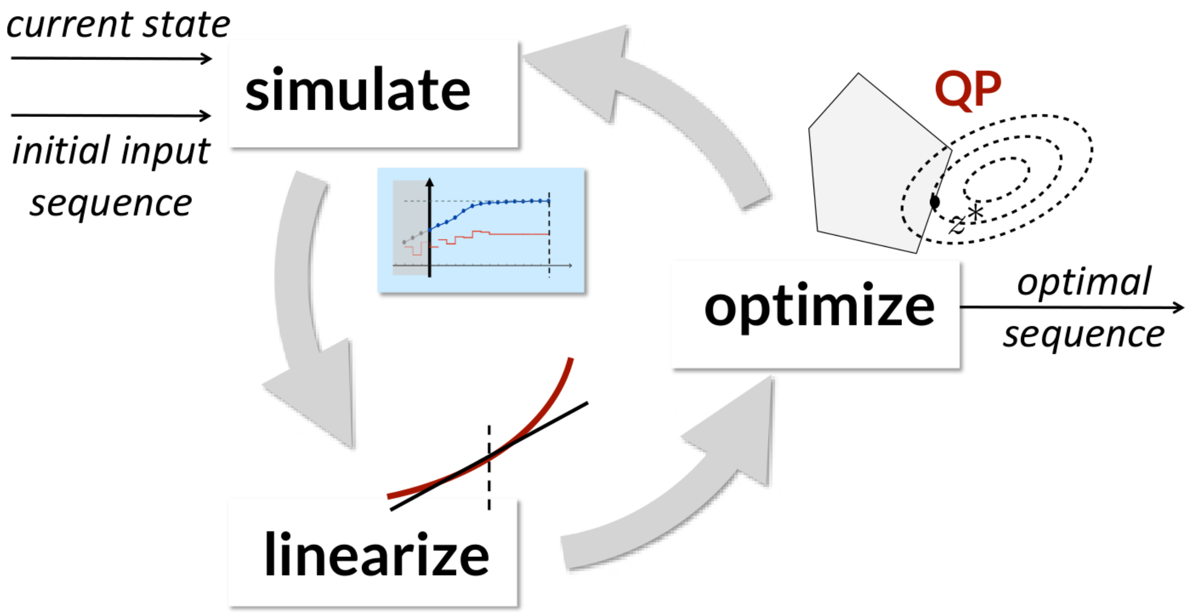
- 用前向欧拉法把线性模型转化为离散形式

$$\dot{\boldsymbol{x}} = \boldsymbol{A}_c\boldsymbol{x} + \boldsymbol{B}_c\boldsymbol{u} + \boldsymbol{g}_c$$

$$\frac{\boldsymbol{x}_{k+1} - \boldsymbol{x}_k}{T_s} = \boldsymbol{A}_c\boldsymbol{x}_k + \boldsymbol{B}_c\boldsymbol{u}_k + \boldsymbol{g}_c$$

$$\boldsymbol{x}_{k+1} = \underline{(\boldsymbol{I} + T_s\boldsymbol{A}_c)}\boldsymbol{x}_k + \underline{T_s\boldsymbol{B}_c}\boldsymbol{u}_k + \underline{T_s\boldsymbol{g}}$$

$$\boldsymbol{x}_{k+1} = \underline{\boldsymbol{A}_k}\boldsymbol{x}_k + \underline{\boldsymbol{B}_k}\boldsymbol{u}_k + \underline{\boldsymbol{g}_k}$$



- 此时可以在线求解线性MPC.

离散线性模型计算

- 线性预测模型 $\begin{cases} \boldsymbol{x}_{k+1} = \boldsymbol{A}\boldsymbol{x}_k + \boldsymbol{B}\boldsymbol{u}_k \\ \boldsymbol{y}_k = \boldsymbol{C}\boldsymbol{x}_k \end{cases}$

- 状态量与输入的关系 $\boldsymbol{x}_k = \boldsymbol{A}^k\boldsymbol{x}_0 + \sum_{j=0}^{k-1}\boldsymbol{A}^j\boldsymbol{B}\boldsymbol{u}_{k-1-j}$

- 状态传播与计算

$$\begin{bmatrix} \boldsymbol{x}_1 \\ \boldsymbol{x}_2 \\ \vdots \\ \boldsymbol{x}_{N-1} \\ \boldsymbol{x}_N \end{bmatrix} = \begin{bmatrix} \boldsymbol{B} & \boldsymbol{0} & \cdots & \boldsymbol{0} \\ \boldsymbol{A}\boldsymbol{B} & \boldsymbol{B} & \cdots & \boldsymbol{0} \\ \vdots & \vdots & \ddots & \vdots \\ \boldsymbol{A}^{N-1}\boldsymbol{B} & \boldsymbol{A}^{N-2}\boldsymbol{B} & \cdots & \boldsymbol{B} \end{bmatrix} \begin{bmatrix} \boldsymbol{u}_0 \\ \boldsymbol{u}_1 \\ \vdots \\ \boldsymbol{u}_{N-1} \end{bmatrix} + \begin{bmatrix} \boldsymbol{A} \\ \boldsymbol{A}^2 \\ \vdots \\ \boldsymbol{A}^N \end{bmatrix} \boldsymbol{x}_0$$

- 线性MPC

- 二次型cost $$J = \boldsymbol{x}_N^T \boldsymbol{P} \boldsymbol{x}_N + \sum_{k=0}^{N-1} \left(\boldsymbol{x}_k^T \boldsymbol{Q} \boldsymbol{x}_k + \boldsymbol{u}_k^T \boldsymbol{R} \boldsymbol{u}_k \right)$$

$$P = P^T \succ 0$$
$$Q = Q^T \succ 0$$
$$R = R^T \succ 0$$

半正定

- 目标：找到最优的控制序列$u_{0:N-1}^*$最小化cost函数 J

$$J = \boldsymbol{x}_0^T \boldsymbol{Q} \boldsymbol{x}_0 + \begin{bmatrix} \boldsymbol{x}_1 \\ \boldsymbol{x}_2 \\ \vdots \\ \boldsymbol{x}_{N-1} \\ \boldsymbol{x}_N \end{bmatrix}^T \begin{bmatrix} \boldsymbol{Q} & \boldsymbol{0} & \boldsymbol{0} & \cdots & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{Q} & \boldsymbol{0} & \cdots & \boldsymbol{0} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ \boldsymbol{0} & \cdots & \boldsymbol{0} & \boldsymbol{Q} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \cdots & \boldsymbol{0} & \boldsymbol{P} \end{bmatrix} \begin{bmatrix} \boldsymbol{x}_1 \\ \boldsymbol{x}_2 \\ \vdots \\ \boldsymbol{x}_{N-1} \\ \boldsymbol{x}_N \end{bmatrix} + \begin{bmatrix} \boldsymbol{u}_0 \\ \boldsymbol{u}_1 \\ \vdots \\ \boldsymbol{u}_{N-1} \end{bmatrix}^T \begin{bmatrix} \boldsymbol{R} & \boldsymbol{0} & \cdots & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{R} & \cdots & \boldsymbol{0} \\ \vdots & \vdots & \ddots & \vdots \\ \boldsymbol{0} & \cdots & \boldsymbol{0} & \boldsymbol{R} \end{bmatrix} \begin{bmatrix} \boldsymbol{u}_0 \\ \boldsymbol{u}_1 \\ \vdots \\ \boldsymbol{u}_{N-1} \end{bmatrix}$$

离散线性模型计算

$$\begin{bmatrix} \boldsymbol{x}_1 \\ \boldsymbol{x}_2 \\ \vdots \\ \boldsymbol{x}_{N-1} \\ \boldsymbol{x}_N \end{bmatrix} = \underbrace{\begin{bmatrix} \boldsymbol{B} & 0 & \cdots & 0 \\ \boldsymbol{AB} & \boldsymbol{B} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ \boldsymbol{A}^{N-1}\boldsymbol{B} & \boldsymbol{A}^{N-2}\boldsymbol{B} & \cdots & \boldsymbol{B} \end{bmatrix}}_{\overline{\boldsymbol{S}}} \underbrace{\begin{bmatrix} \boldsymbol{u}_0 \\ \boldsymbol{u}_1 \\ \vdots \\ \boldsymbol{u}_{N-1} \end{bmatrix}}_{\boldsymbol{z}} + \underbrace{\begin{bmatrix} \boldsymbol{A} \\ \boldsymbol{A}^2 \\ \vdots \\ \boldsymbol{A}^N \end{bmatrix}}_{\overline{\boldsymbol{T}}} \boldsymbol{x}_0$$

$$J(\boldsymbol{z}, \boldsymbol{x}_0) = \boldsymbol{x}_0^T \boldsymbol{Q} \boldsymbol{x}_0 + \begin{bmatrix} \boldsymbol{x}_1 \\ \boldsymbol{x}_2 \\ \vdots \\ \boldsymbol{x}_{N-1} \\ \boldsymbol{x}_N \end{bmatrix}^T \overbrace{\begin{bmatrix} \boldsymbol{Q} & 0 & 0 & \cdots & 0 \\ 0 & \boldsymbol{Q} & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & \cdots & 0 & \boldsymbol{Q} & 0 \\ 0 & 0 & \cdots & 0 & \boldsymbol{P} \end{bmatrix}}^{\overline{\boldsymbol{Q}}} \begin{bmatrix} \boldsymbol{x}_1 \\ \boldsymbol{x}_2 \\ \vdots \\ \boldsymbol{x}_{N-1} \\ \boldsymbol{x}_N \end{bmatrix} + \begin{bmatrix} \boldsymbol{u}_0 \\ \boldsymbol{u}_1 \\ \vdots \\ \boldsymbol{u}_{N-1} \end{bmatrix}^T \overbrace{\begin{bmatrix} \boldsymbol{R} & 0 & \cdots & 0 \\ 0 & \boldsymbol{R} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & \cdots & 0 & \boldsymbol{R} \end{bmatrix}}^{\overline{\boldsymbol{R}}} \begin{bmatrix} \boldsymbol{u}_0 \\ \boldsymbol{u}_1 \\ \vdots \\ \boldsymbol{u}_{N-1} \end{bmatrix}$$

$$\begin{aligned} J(\boldsymbol{z}, \boldsymbol{x}_0) &= \left(\overline{\boldsymbol{S}}\boldsymbol{z} + \overline{\boldsymbol{T}}\boldsymbol{x}_0\right)^T \overline{\boldsymbol{Q}} \left(\overline{\boldsymbol{S}}\boldsymbol{z} + \overline{\boldsymbol{T}}\boldsymbol{x}_0\right) + \boldsymbol{z}^T \overline{\boldsymbol{R}} \boldsymbol{z} + \boldsymbol{x}_0^T \boldsymbol{Q} \boldsymbol{x}_0 \\ &= \frac{1}{2} \boldsymbol{z}^T \underbrace{2\left(\overline{\boldsymbol{R}} + \overline{\boldsymbol{S}}^T \overline{\boldsymbol{Q}}\, \overline{\boldsymbol{S}}\right)}_{\boldsymbol{H}} \boldsymbol{z} + \boldsymbol{x}_0^{\mathrm{T}} \underbrace{2\overline{\boldsymbol{T}}^T \overline{\boldsymbol{Q}}\, \overline{\boldsymbol{S}}}_{\boldsymbol{F}} \boldsymbol{z} + \frac{1}{2} \boldsymbol{x}_0^T \underbrace{2\left(\boldsymbol{Q} + \overline{\boldsymbol{T}}^T \overline{\boldsymbol{Q}}\, \overline{\boldsymbol{T}}\right)}_{\boldsymbol{Y}} \boldsymbol{x}_0 \end{aligned}$$

离散线性模型计算

$$J(\boldsymbol{z},\boldsymbol{x}_0) = \frac{1}{2}\boldsymbol{z}^T \underbrace{2\left(\overline{\boldsymbol{R}} + \overline{\boldsymbol{S}}^T\overline{\boldsymbol{Q}}\,\overline{\boldsymbol{S}}\right)}_{\boldsymbol{H}}\boldsymbol{z} + \boldsymbol{x}_0^T \underbrace{2\overline{\boldsymbol{T}}^T\overline{\boldsymbol{Q}}\,\overline{\boldsymbol{S}}}_{\boldsymbol{F}}\boldsymbol{z} + \frac{1}{2}\boldsymbol{x}_0^T \underbrace{2\left(\boldsymbol{Q} + \overline{\boldsymbol{T}}^T\overline{\boldsymbol{Q}}\,\overline{\boldsymbol{T}}\right)}_{\boldsymbol{Y}}\boldsymbol{x}_0$$

- MPC的紧凑形式

$$J(\boldsymbol{z},\boldsymbol{x}_0) = \frac{1}{2}\boldsymbol{z}^T\boldsymbol{H}\boldsymbol{z} + \boldsymbol{x}_0^T\boldsymbol{F}\boldsymbol{z} + \frac{1}{2}\boldsymbol{x}_0^T\boldsymbol{Y}\boldsymbol{x}_0 \qquad \boldsymbol{z} = \begin{bmatrix}\boldsymbol{u}_0\\ \boldsymbol{u}_1\\ \vdots\\ \boldsymbol{u}_{N-1}\end{bmatrix}$$

目标即是找到梯度的为零的解

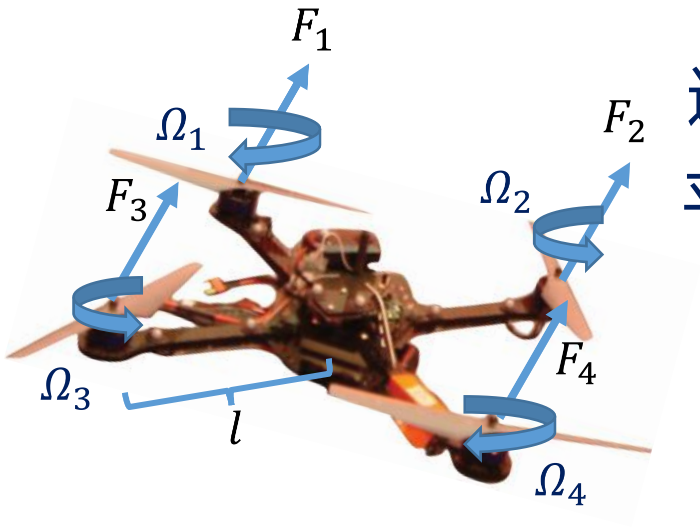
$$\nabla_{\boldsymbol{z}} J(\boldsymbol{z},\boldsymbol{x}_0) = \boldsymbol{H}\boldsymbol{z} + \boldsymbol{F}^T\boldsymbol{x}_0 = 0 \rightarrow \boldsymbol{z}^* = -\underline{\boldsymbol{H}^{-1}}\boldsymbol{F}^T\boldsymbol{x}_0$$ (解序列“batch” solution)

无约束MPC= 线性反馈控制(linear state-feedback)

Wang, Yang, and Stephen Boyd. "Fast model predictive control using online optimization." IEEE Transactions on control systems technology 18.2 (2009): 267-278.

线性模型计算实例

- 例子：选择最佳系统输入即总推力U_1与x轴力矩U_2



这里采用二维简化模型 $\dot{\boldsymbol{x}} = f_c(\boldsymbol{x}, u) = \begin{pmatrix} \dot{z} \\ -g + \frac{1}{m}U_1 \cos\phi \\ \dot{y} \\ -\frac{1}{m}U_1 \sin\phi \\ \dot{\phi} \\ \frac{1}{I_x}U_2 \end{pmatrix}$

平衡悬停态 $\theta = \dot{\theta} = \psi = \dot{\psi} \sim 0$

系统状态$\boldsymbol{x} = [z, \dot{z}, y, \dot{y}\ \phi, \dot{\phi}]$

系统输入$\boldsymbol{u} = [U_1, U_2]$

线性模型计算实例

- 线性模型离散化

$$\dot{\boldsymbol{x}} = f_c(\boldsymbol{x}, u) = \begin{pmatrix} \dot{z} \\ -g + \frac{1}{m}U_1\cos\phi \\ \dot{y} \\ -\frac{1}{m}U_1\sin\phi \\ \dot{\phi} \\ \frac{1}{I_x}U_2 \end{pmatrix} \xrightarrow{\text{前向欧拉法}} \boldsymbol{x}_{t+1} = f(\boldsymbol{x}_t, u_t) = \begin{pmatrix} \boldsymbol{x}_t^1 + h\boldsymbol{x}_t^2 \\ \boldsymbol{x}_t^2 + h\left(-g + \frac{1}{m}u_t^1\cos\boldsymbol{x}_t^5\right) \\ \boldsymbol{x}_t^3 + h\boldsymbol{x}_t^4 \\ \boldsymbol{x}_t^4 - \frac{h}{m}u_t^1\sin\boldsymbol{x}_t^5 \\ \boldsymbol{x}_t^5 + h\boldsymbol{x}_t^6 \\ \boldsymbol{x}_t^6 + \frac{h}{I_x}u_t^2 \end{pmatrix}$$

在工作点$(\boldsymbol{x}_t^{ref}, u_t^{ref})$处线性展开

$$\boldsymbol{x}_{t+1} = f(\boldsymbol{x}_t,\ u_t) \approx f(\boldsymbol{x}_t^{ref},\ u_t^{ref}) + \widetilde{A}_t(\boldsymbol{x}_t - \boldsymbol{x}_t^{ref}) + \widetilde{B}_t(u_t - u_t^{ref})$$

$$\widetilde{A}_t = \frac{\partial f(\boldsymbol{x},u)}{\partial \boldsymbol{x}}\bigg|_{(\boldsymbol{x}_t^{ref},\ u_t^{ref})} \Longrightarrow \widetilde{A}_t = \begin{bmatrix} 1 & h & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & -\frac{h}{m}u_t^{ref,1}\sin\boldsymbol{x}_t^{ref,5} & 0 \\ 0 & 0 & 1 & h & 0 & 0 \\ 0 & 0 & 0 & 1 & -\frac{h}{m}u_t^{ref,1}\cos\boldsymbol{x}_t^{ref,5} & 0 \\ 0 & 0 & 0 & 0 & 1 & h \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\widetilde{B}_t = \frac{\partial f(\boldsymbol{x},u)}{\partial u}\bigg|_{(\boldsymbol{x}_t^{ref},\ u_t^{ref})} \Longrightarrow \widetilde{B}_t = \begin{bmatrix} 0 & 0 \\ \frac{h}{m}\cos\boldsymbol{x}_t^{ref,5} & 0 \\ 0 & 0 \\ -\frac{h}{m}\sin\boldsymbol{x}_t^{ref,5} & 0 \\ 0 & 0 \\ 0 & \frac{h}{I_x} \end{bmatrix}$$

线性模型计算实例

- 线性模型离散化

$$A_t = \begin{bmatrix} 1 & h & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & -\frac{h}{m}\overline{u}_t^1 \sin \overline{\boldsymbol{x}}_t^5 & 0 \\ 0 & 0 & 1 & h & 0 & 0 \\ 0 & 0 & 0 & 1 & -\frac{h}{m}\overline{u}_t^1 \cos \overline{\boldsymbol{x}}_t^5 & 0 \\ 0 & 0 & 0 & 0 & 1 & h \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \qquad B_t = \begin{bmatrix} 0 & 0 \\ \frac{h}{m}\cos \overline{\boldsymbol{x}}_t^5 & 0 \\ 0 & 0 \\ -\frac{h}{m}\sin \overline{\boldsymbol{x}}_t^5 & 0 \\ 0 & 0 \\ 0 & \frac{h}{I_x} \end{bmatrix}$$

系统状态$\boldsymbol{x} = [z, \dot{z}, y, \dot{y}\ \phi, \dot{\phi}\ ]$

系统输入$\boldsymbol{u} = [U_1, U_2\ ]$

$$\boldsymbol{g}_t = f(\boldsymbol{x}_t^{ref}, u_t^{ref}) - A_t\, \boldsymbol{x}_t^{ref} - B_t u_t^{ref}$$

$$\boldsymbol{x}_{t+1} = \boldsymbol{A}_t \boldsymbol{x}_t + \boldsymbol{B}_t \boldsymbol{u}_t + \boldsymbol{g}_t$$

- 最小化的累加和

$$\left(p_x - x_{ref}\right)^2 + \left(p_y - y_{ref}\right)^2 + w_v \Delta a^2 + w_\delta \Delta\delta^2$$ →二次型

- 增广模型

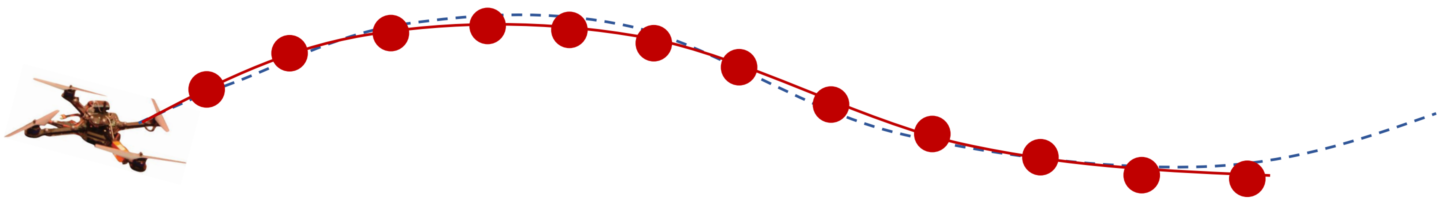
$$\begin{bmatrix} \boldsymbol{x} \\ \boldsymbol{u} \end{bmatrix}_{k+1} = \begin{bmatrix} \boldsymbol{A} & \boldsymbol{B} \\ \boldsymbol{0} & \boldsymbol{I} \end{bmatrix} \begin{bmatrix} \boldsymbol{x} \\ \boldsymbol{u} \end{bmatrix}_k + \begin{bmatrix} \boldsymbol{B} \\ \boldsymbol{I} \end{bmatrix} \Delta \boldsymbol{u}_k \qquad z = \begin{bmatrix} \boldsymbol{u}_0 \\ \boldsymbol{u}_1 \\ \vdots \\ \boldsymbol{u}_{N-1} \end{bmatrix} \text{ or } z = \begin{bmatrix} \Delta\boldsymbol{u}_0 \\ \Delta\boldsymbol{u}_1 \\ \vdots \\ \Delta\boldsymbol{u}_{N-1} \end{bmatrix}$$

- 约束形式

$$a_{\min} \le a \le a_{\max}$$
$$v_{\min} \le v \le v_{\max}$$

→ 线性约束二次型（Linear constrained QP formulation）

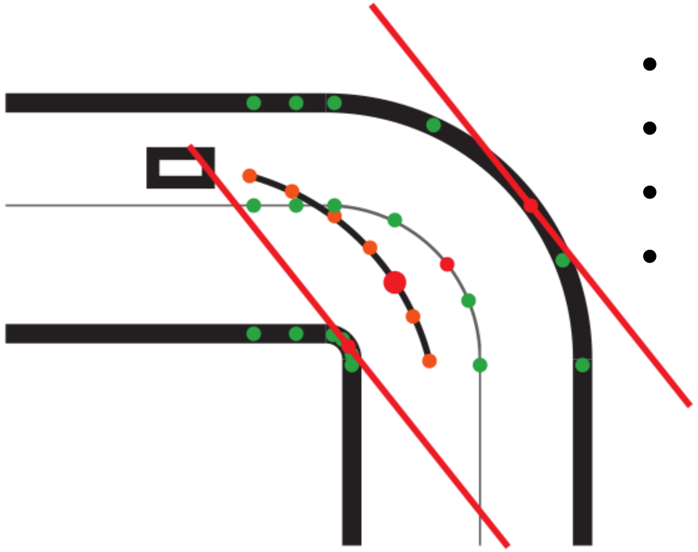
线性模型计算实例



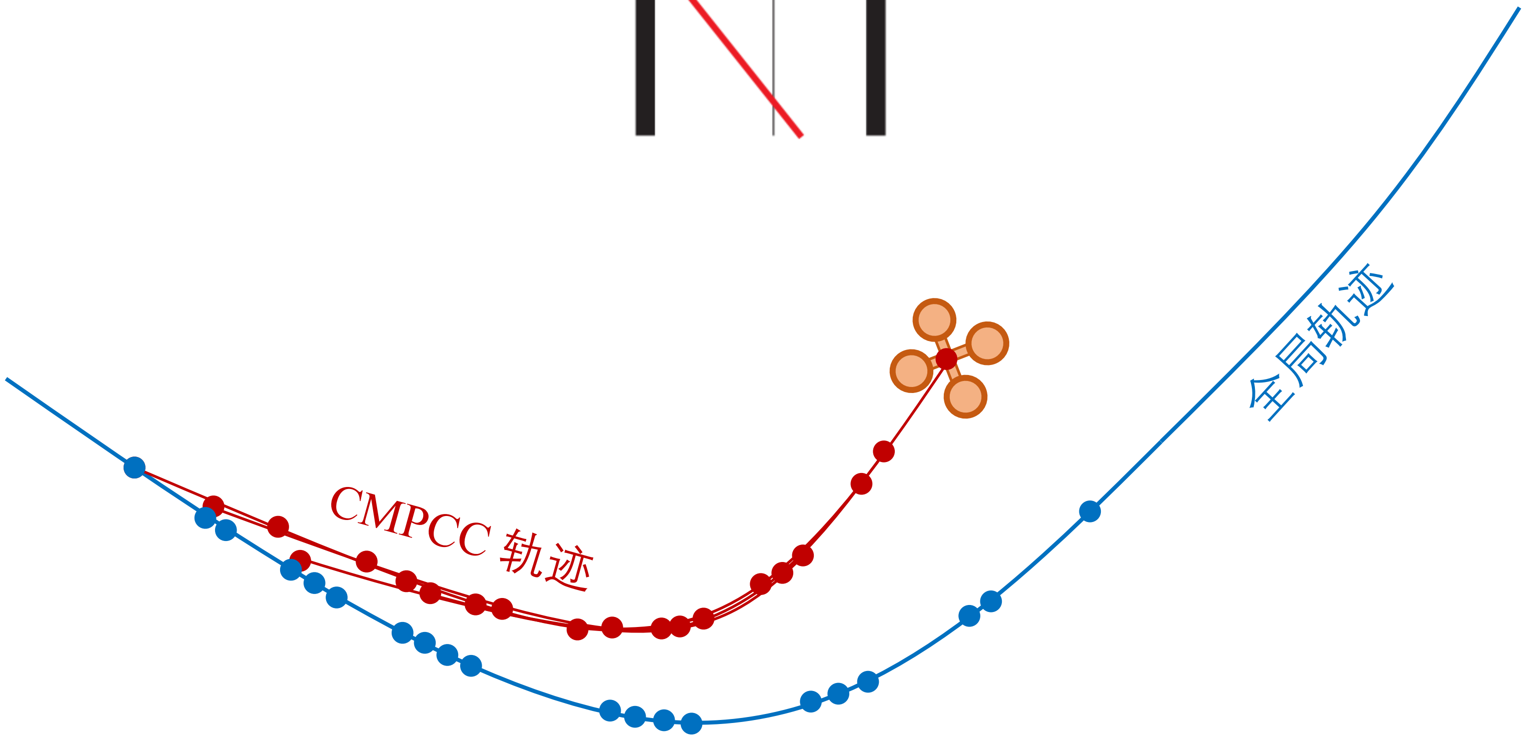
The chicken or the egg

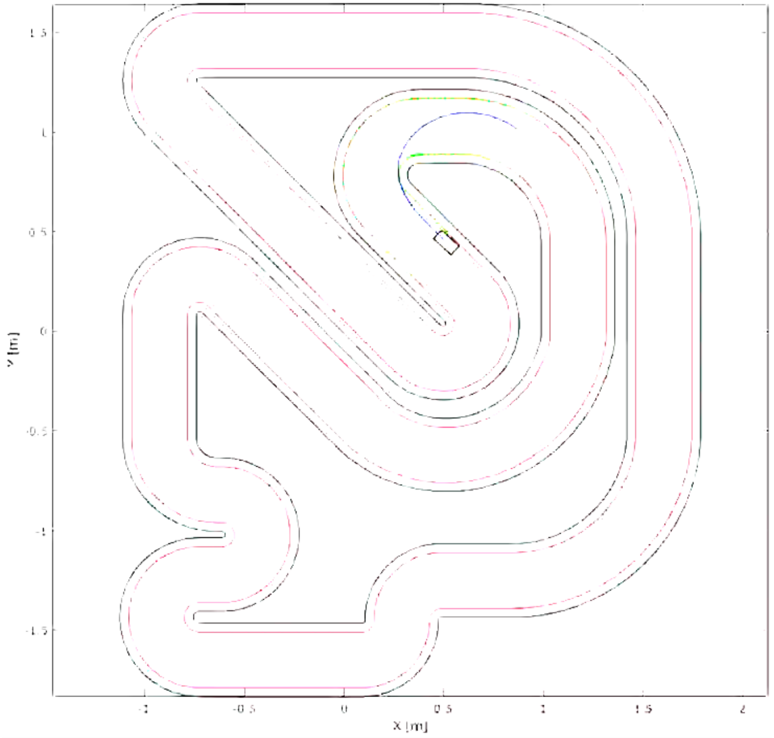
- 一些工程技巧：
  - 根据参考轨迹人为给定
  - Warm-start：在上一次求解处对模型线性化 u_{k-1}^*, x_{k-1}^*

CMPCC



- 平衡跟踪进度和执行进度.
- 走廊约束保证安全.
- 终端速度约束保证可行.
- QP方法让问题在机载电脑上求解时间少于5ms.





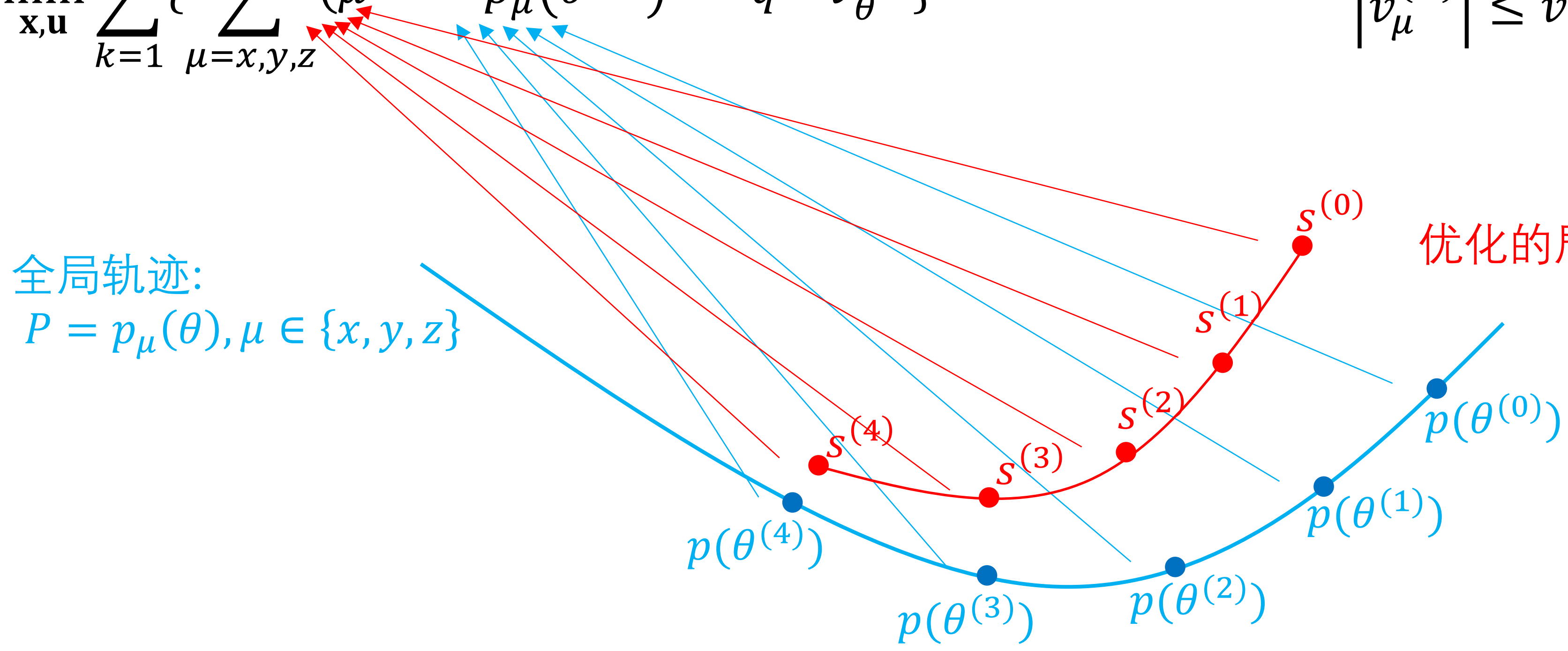
◆ 目标函数

状态：$\mathbf{x}^{(k)} = [x, v_x, a_x, y, v_y, a_y, z, v_z, a_z, \theta, v_\theta, a_\theta]^T$

输入：$\mathbf{u}^{(k)} = [j_x, j_y, j_z, j_\theta]^T$

$$J = \min_{\mathbf{x},\mathbf{u}} \sum_{k=1}^{N} \{ \sum_{\mu=x,y,z} (\mu^{(k)} - p_\mu(\theta^{(k)})^2 - q \cdot v_\theta^{(k)} \}$$

$$\begin{aligned} \text{s.t. } & \mathbf{x}^{(k+1)} = \mathbf{A_d}\mathbf{x}^{(k)} + \mathbf{B_d}\mathbf{u}^{(k)}, k = 1,2,3,\dots,N-1 \\ & \mathbf{x}_l \le \mathbf{x}^k \le \mathbf{x}_u, k = 1,2,3,\dots,N-1 \\ & \mathbf{u}_l \le \mathbf{u}^k \le \mathbf{u}_u, k = 1,2,3,\dots,N-1 \\ & \mathbf{C}^k \cdot [x^k, y^k, z^k]^T \le \mathbf{b}^k, k = 1,2,3,\dots,N-1 \\ & \left| v_\mu^{(N)} \right| \le v_{t\mu}, \mu = x,y,z \end{aligned}$$



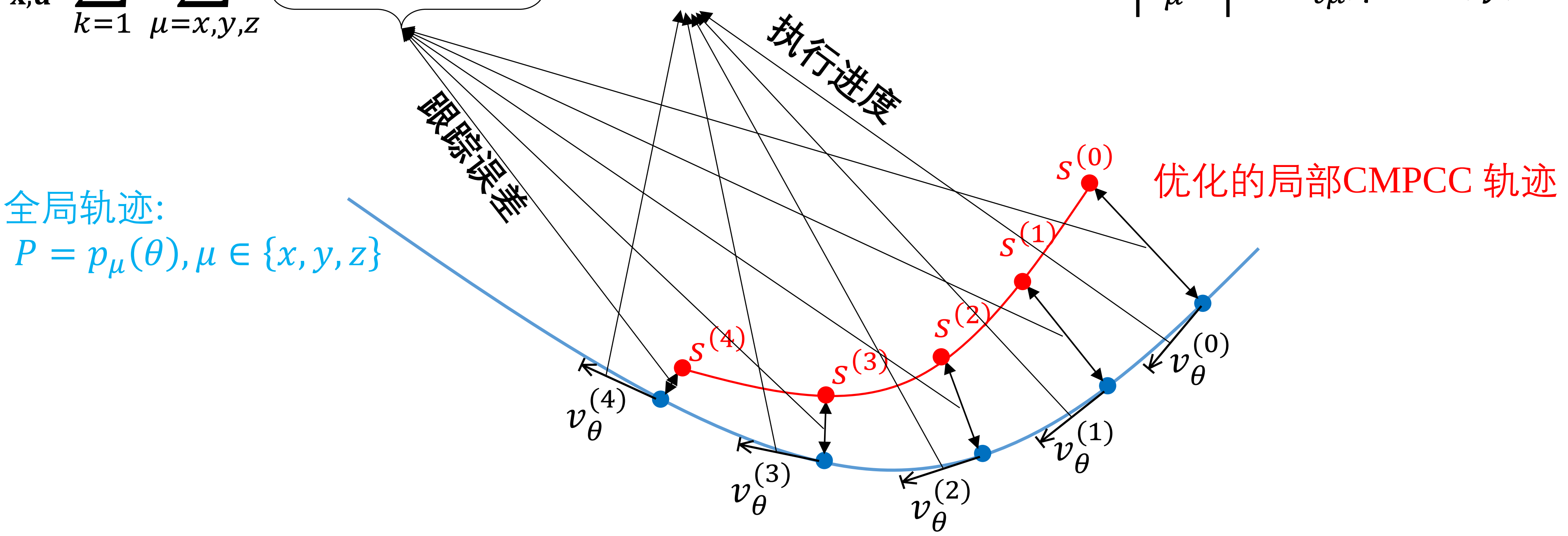
◆ 目标函数

状态: $\mathbf{x}^{(k)} = [x, v_x, a_x, y, v_y, a_y, z, v_z, a_z, \theta, v_\theta, a_\theta]^T$

输入: $\mathbf{u}^{(k)} = [j_x, j_y, j_z, j_\theta]^T$

$$J = \min_{\mathbf{x},\mathbf{u}} \sum_{k=1}^{N} \{ \sum_{\mu=x,y,z} \underbrace{(\mu^{(k)} - p_\mu(\theta^{(k)}))^2}_{跟踪误差} - q \cdot \underbrace{v_\theta^{(k)}}_{执行进度} \}$$

$$\begin{aligned} \text{s.t. } & \mathbf{x}^{(k+1)} = \mathbf{A_d}\mathbf{x}^{(k)} + \mathbf{B_d}\mathbf{u}^{(k)}, k = 1,2,3,\dots,N-1 \\ & \mathbf{x}_l \le \mathbf{x}^k \le \mathbf{x}_u, k = 1,2,3,\dots,N-1 \\ & \mathbf{u}_l \le \mathbf{u}^k \le \mathbf{u}_u, k = 1,2,3,\dots,N-1 \\ & \mathbf{C}^k \cdot [x^k, y^k, z^k]^T \le \mathbf{b}^k, k = 1,2,3,\dots,N-1 \\ & \left| v_\mu^{(N)} \right| \le v_{t\mu}, \mu = x, y, z \end{aligned}$$



◆ 终端速度约束

限制终端速度可确保轨迹可行性，同时大大缩短规划的时间。

$$\mathbf{s.t.}\ \mathbf{x}^{(k+1)} = \mathbf{A_d}\mathbf{x}^{(k)} + \mathbf{B_d}\mathbf{u}^{(k)}, k = 1,2,3,\dots,N-1$$
$$\mathbf{x}_l \le \mathbf{x}^k \le \mathbf{x}_u, k = 1,2,3,\dots,N-1$$
$$\mathbf{u}_l \le \mathbf{u}^k \le \mathbf{u}_u, k = 1,2,3,\dots,N-1$$
$$\mathbf{C}^k \cdot [\boldsymbol{x}^k, y^k, z^k]^T \le \mathbf{b}^k, k = 1,2,3,\dots,N-1$$
$$\left|v_\mu^{(N)}\right| \le v_{t\mu}, \mu = x, y, z$$