

自动驾驶控制算法第八讲

$$u = -K_{err} + \delta_f$$

k : 三四五讲 e_{ir} : 四, 七讲 δ_f : 六讲

综合起来：八讲 \Rightarrow 完整算法

算法输入

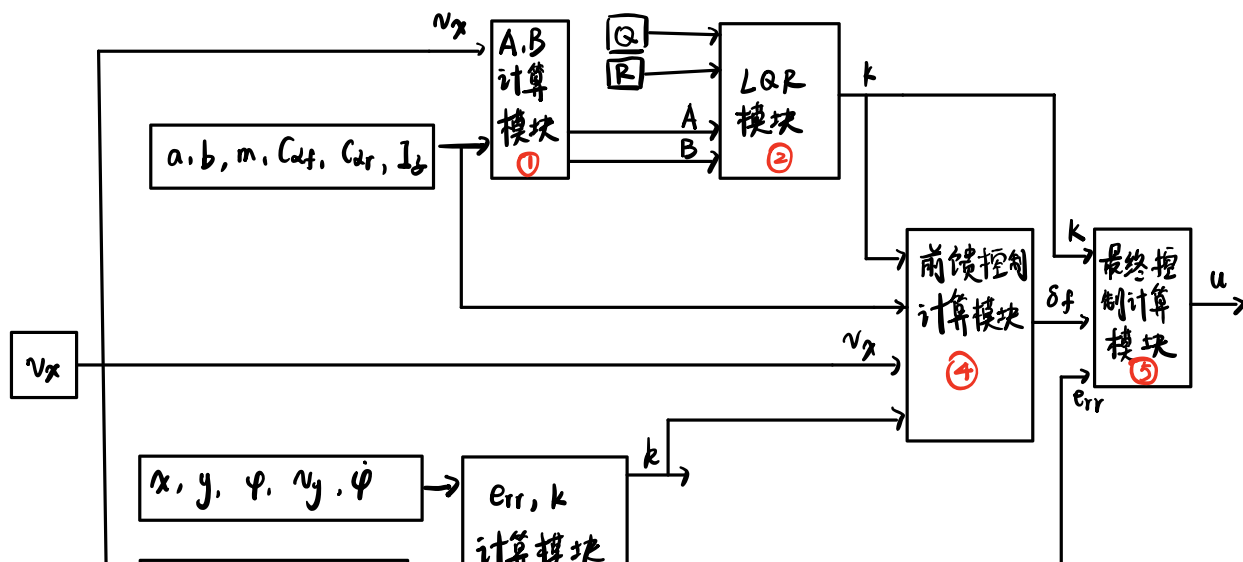
1. 整车参数: $a, b, C_{df}, C_{dr}, m, I_z$
2. 车辆位置与状态: $x, y, \varphi, v_x, v_y, \dot{\varphi}$
3. 规划轨迹点:
$$\begin{pmatrix} x_r \\ y_r \\ \theta_r \\ k_r \end{pmatrix} = \begin{pmatrix} x_1 & x_2 & & \\ y_1 & y_2 & \dots & \\ \theta_1 & \theta_2 & & \\ k_1 & k_2 & & \end{pmatrix}$$

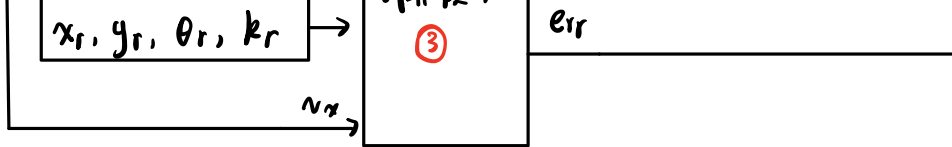
算法输出: $u = -k e_{1r} + \delta f$

过程

1. 整车参数 + $v_x \Rightarrow A, B \xrightarrow[\text{LQR}]{Q, R} K$
 k (投影的曲率)
2. 车辆位置与状态 + 规划轨迹点 $\Rightarrow e_{rr}$
3. 1步的 K + 2步的 k + 整车参数 + $v_x \Rightarrow \delta f$
- ④ 1步的 k + 2步的 e_{rr} + 3步的 $\delta f \Rightarrow u = -Kx + \delta f$

流程图





具体算法:

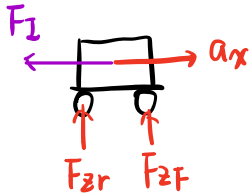
① A, B 计算模块

$$A = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & \frac{C_{df} + C_{dr}}{m v_x} & -\frac{C_{df} + C_{dr}}{m} & \frac{a C_{df} + b C_{dr}}{m v_x} \\ 0 & 0 & 0 & 1 \\ 0 & \frac{a C_{df} - b C_{dr}}{m v_x} & -\frac{C_{df} + C_{dr}}{I_b} & \frac{a^2 C_{df} + b^2 C_{dr}}{I_b v_x} \end{pmatrix} \quad B = \begin{pmatrix} 0 \\ -\frac{C_{df}}{m} \\ 0 \\ -\frac{a C_{df}}{I_b} \end{pmatrix}$$

注意要考虑当 $v_x = 0$ 时的奇异性

② LQR 模块

$\therefore A, B$ 只与整车参数和 v_x 有关, 整车参数近似认为不变



仍可用自行车模型
只是侧偏刚度将变化



此时自行车模型不再适用

A, B 只与 v_x 有关 $K = lqr(A, B, Q, R)$ (或 $dlqr(\bar{A}, \bar{B}, Q, R)$)

每一个 v_x 都有唯一的一个 K 与之对应

\therefore 可离线计算出 v_x 与 K 的对应表, 实际应用不求解 Riccati 方程, 直接查表

表	0.01	0.02	0.03	...	50	50m/s 或 180 km/h
	k_1	k_2	k_3		k_n	



优点: 速度大大加快

缺点: 耗费存储空间 (空间换时间)

③ err. k 计算模块

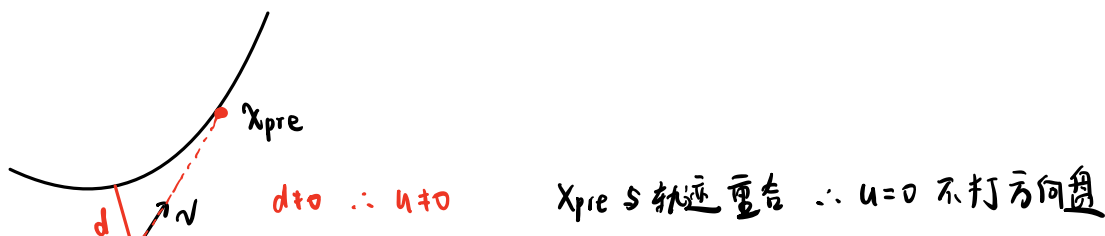
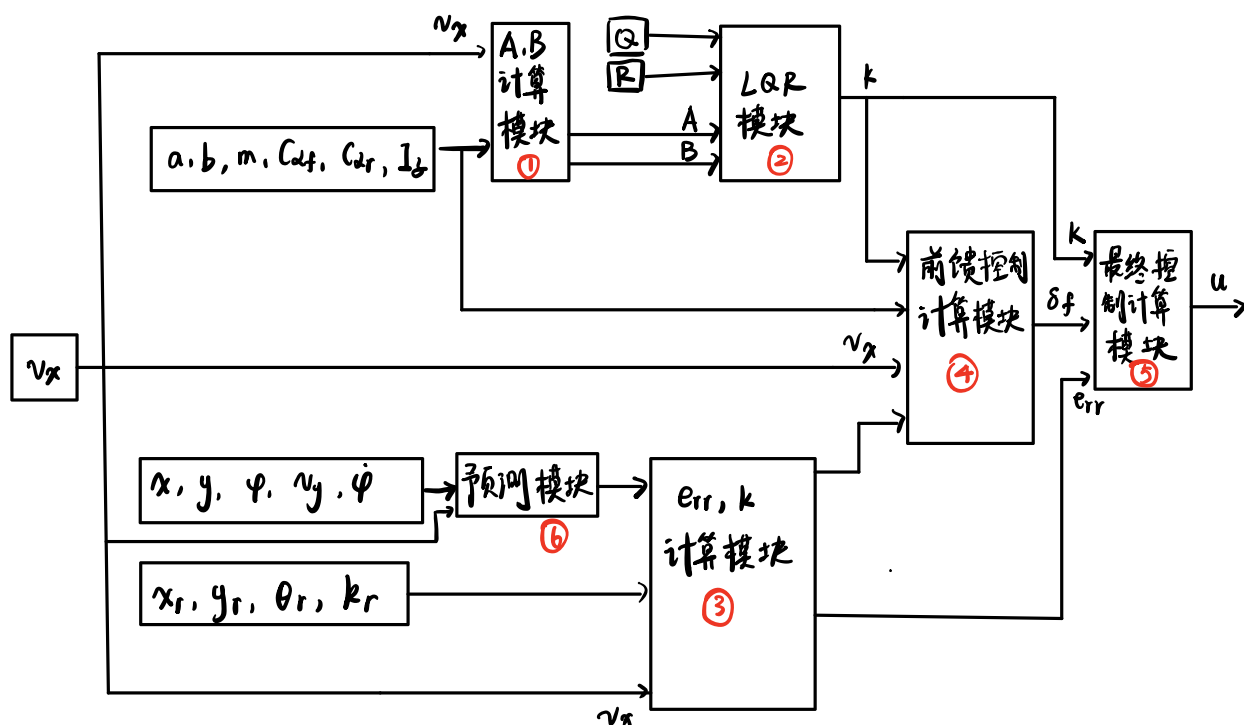
1.
$$\begin{pmatrix} x_r \\ y_r \\ \theta_r \\ k_r \end{pmatrix} = \begin{pmatrix} x_1 & x_2 \\ y_1 & y_2 & \dots \\ \theta_1 & \theta_2 \\ k_1 & k_2 \end{pmatrix}$$
2. 遍历 $\begin{pmatrix} x_r \\ y_r \end{pmatrix}$ 找到与 $\begin{pmatrix} x \\ y \end{pmatrix}$ 最近的规划点, 该点的序列记为 d_{min}
3. $\vec{c} = \begin{pmatrix} \cos(\theta_{dmin}) \\ \sin(\theta_{dmin}) \end{pmatrix} \quad \vec{n} = \begin{pmatrix} -\sin(\theta_{dmin}) \\ \cos(\theta_{dmin}) \end{pmatrix}$
4. $d_err = \begin{pmatrix} x - x_{dmin} \\ y - y_{dmin} \end{pmatrix}$
5. $e_d = \vec{n}^T \cdot d_err$
6. $e_s = \vec{c}^T \cdot d_err$
7. $\theta_r = \theta_{min} \text{ (apollo)} \quad \theta_r = \theta_{min} + k_{dmin} \cdot e_s \text{ (我)}$
8. $\dot{e}_d = |\vec{v}| \sin(\theta - \theta_r) = v_y \cos(\varphi - \theta_r) + v_x \sin(\varphi - \theta_r)$
9. $e_\varphi = \varphi - \theta_r$
10. $\dot{s} = \frac{v \cos(\theta - \theta_r)}{1 - k_{dmin} \cdot e_d} = \frac{v_x \cos(\varphi - \theta_r) - v_y \sin(\varphi - \theta_r)}{1 - k_{dmin} e_d}$
11. $\dot{e}_\varphi = \dot{\varphi} - k_{dmin} \cdot \dot{s}$
12. $k_r = k_{dmin}$
13. 输出 $e_d, \dot{e}_d, e_\varphi, \dot{e}_\varphi, k_r$

④ 前馈控制计算模块

1. $k_3 = K(3)$
2. $\delta_f = k \left[a + b - b k_3 - \frac{m v_x^2}{a+b} \left(\frac{b}{C_{df}} + \frac{a}{C_{dr}} k_3 - \frac{a}{C_{dr}} \right) \right]$

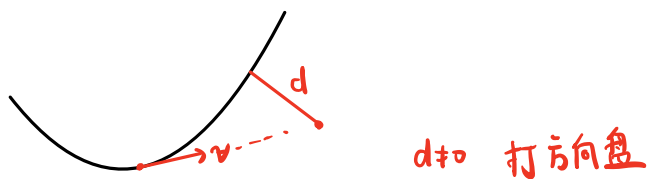
⑤ 最终控制计算模块

$$\delta = -k_{err} + \delta_f$$



人开车：人知道未来的路径规划是什么，人不会打方向盘

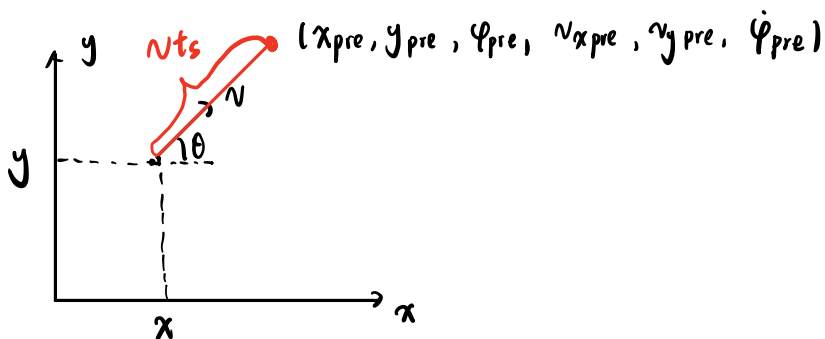
算法开车： $err \neq 0$ ，算法会打方向盘



人开车：人知道未来的路径规划不是直线，人会动方向盘

算法开车： $err=0$ $\delta_f \approx 0$ 算法不会动方向盘

算法控制具有滞后性 \therefore 加上预测模块



⑥ 预测模块

预测时间记为 t_s

$$\text{有 } x_{pre} = x + v t_s \cos \theta = x + v t_s \cos(\beta + \varphi) = x + v_x t_s \cos \varphi - v_y t_s \sin \varphi$$

$$y_{pre} = y + v t_s \sin \theta = y + v_y t_s \cos \varphi + v_x t_s \sin \varphi$$

$$\varphi_{pre} = \varphi + \dot{\varphi} t_s$$

$$v_{xpre} = v_x$$

$$v_{ypre} = v_y$$

$$\dot{\varphi}_{pre} = \dot{\varphi}$$