

# 机器人学导论

## 第八讲 路径规划

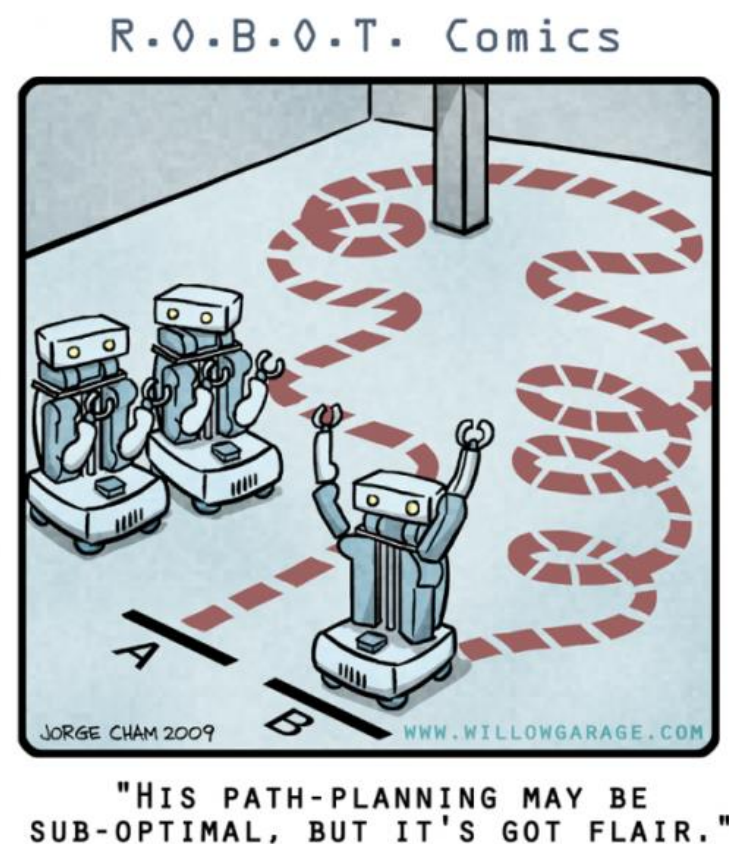
黄之峰

广东工业大学

自动化学院

# 主要内容：

1. 机器人的路径及轨迹的相关定义
2. 路径规划算法概述
3. 关节空间与笛卡尔空间的区别
4. RRT路径规划算法
5. 设计项目III-3自由度机器人路径规划实践

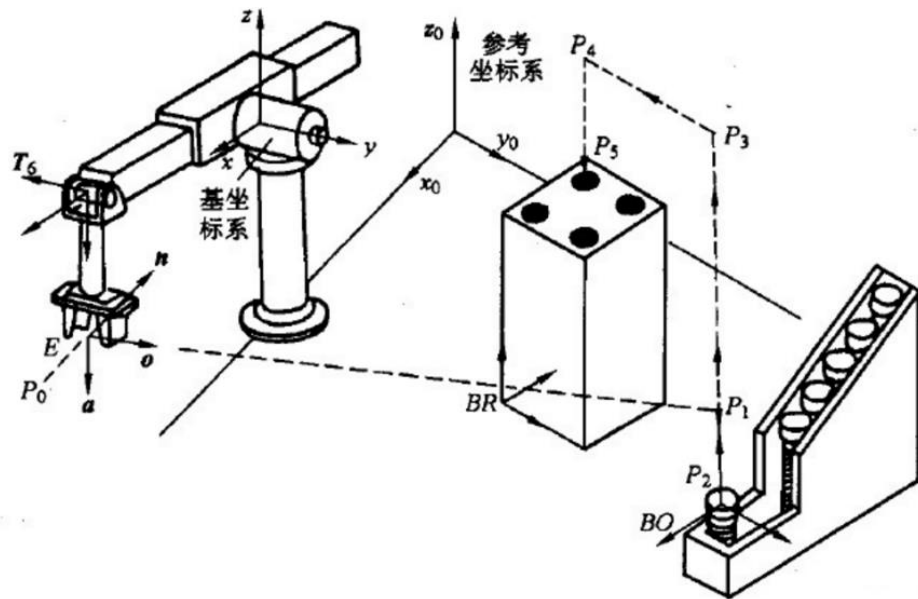


# 第八讲 1-机器人路径及轨迹规划的基本问题

基本问题：如何控制机器人从起始点运动到目标点，如何进行加减速（轻拿轻放），有时还应该考虑经过/避开某些中间点。

答案A：可以通过拖动  
/面板示教方法实现。

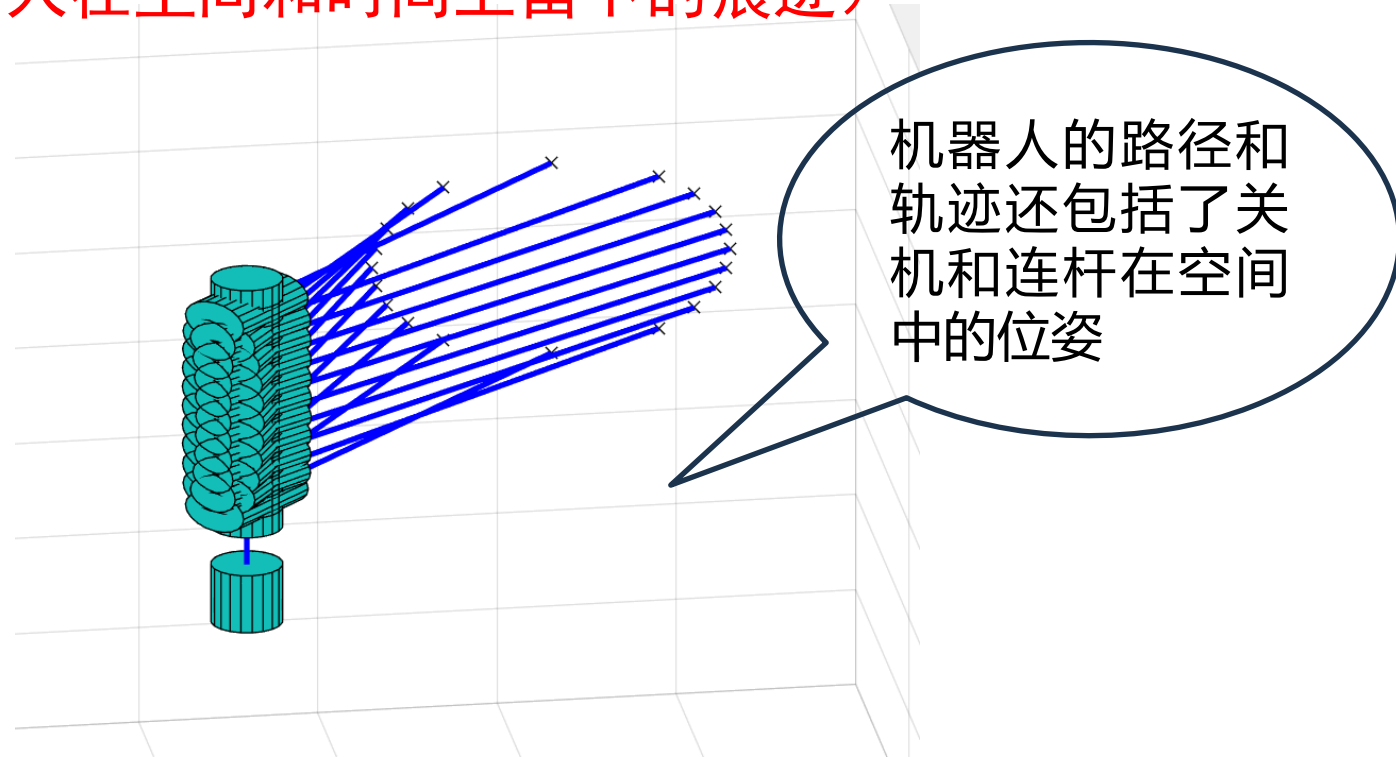
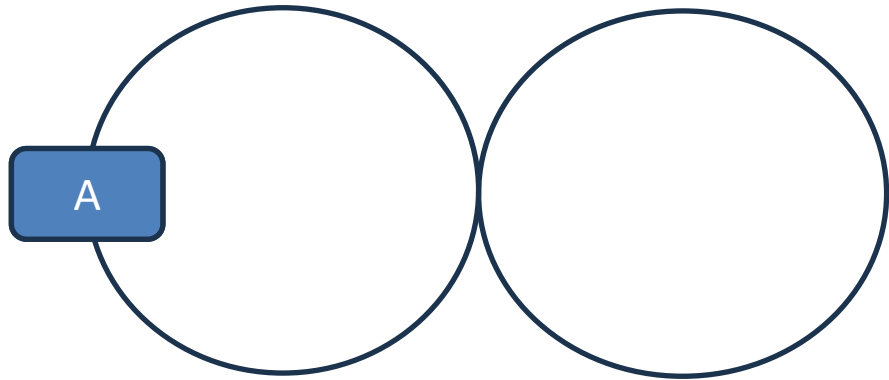
可行，但是  
不够好，不  
够自动



# 第八讲 1-机器人的路径及轨迹的相关定义

**路径 (Path)**：空间位移序列，机器人位形的一个特定序列，不考虑机器人位形的时间因素。（机器人在空间留下的痕迹）

**轨迹 (Trajectory)**：带时间约束的位移序列，与何时到达路径中的每个部分有关，**强调时间性，依赖于速度和加速度。**（机器人在空间和时间上留下的痕迹）



## 第八讲 1-机器人的路径及轨迹的相关定义

**路径规划**：指在地图上生成一条连接起点和终点的路径曲线，该路径曲线**不会与地图中的障碍物**相交，且均在可行区域，路径曲线Path可以用离散的点序列表示如下：

$$\mathbf{Path} = \{(x_k, y_k) \mid k = 1, 2, \dots, n\} \quad (1)$$

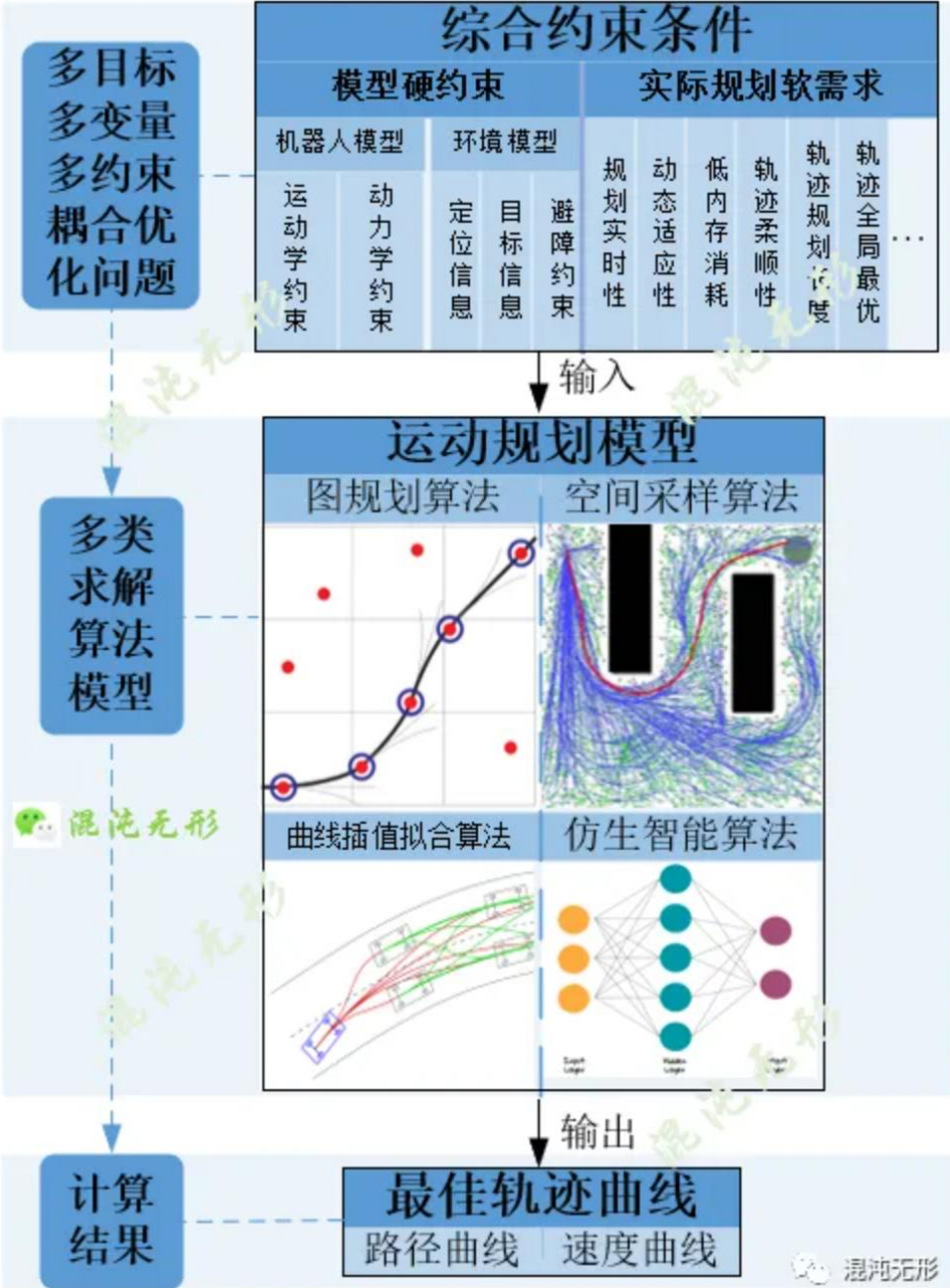
**轨迹规划**：顾名思义就是在地图上生成一条连接起点和终点的轨迹曲线，而轨迹曲线是**路径曲线和速度曲线相耦合的复合曲线**，换句话说，就是轨迹曲线Traj包含了位置、速度和时间等信息，离散化后可表示为：**(轨迹曲线=路径曲线+速度曲线)**

$$\mathbf{Traj} = \{(x_t, y_t, v_t, w_t) \mid t = 1, 2, \dots, n\} \quad (2)$$

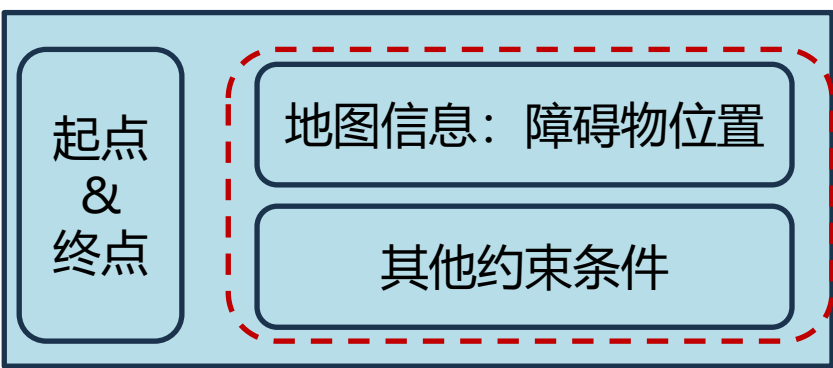
**运动规划**：狭义上与轨迹规划相近，但根据机器人不同，还需要进一步把期望轨迹转换为机器人关节/执行器的期望轨迹（力矩曲线），这一轨迹还需要补偿上机器人本身动力学模型所带来的特性（惯性，重力、哥氏力等）。



# 第八讲 2-路径规划算法概述



输入



规划算法

输出

路点

需要注意的是，对于机械臂来说，起点，终点以及路点，不一定是末端位姿，大多数时候还包括了各个关节角的序列

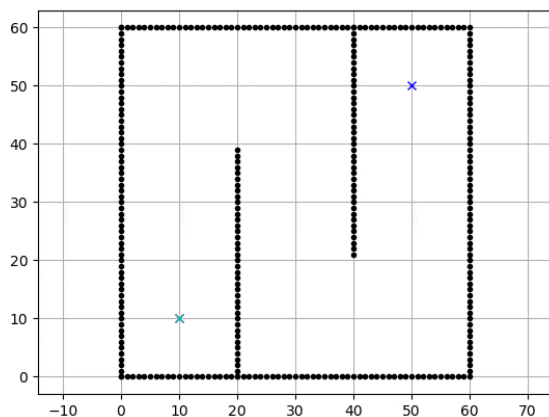
# 第八讲 2-路径规划算法概述

## 1, 图规划算法

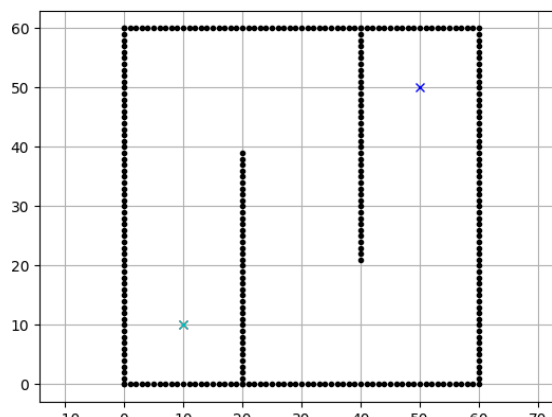
a) 图搜索算法——计算所有途径节点的路径的成本值，并选择成本值最低的一条作为结果。

b) BUG类算法——遇到障碍物就沿着障碍物边缘运动，实际生成的路径曲线与障碍物边缘曲线形态有着密切关系，可以近似认为是障碍物边界向外缩放后的曲线。

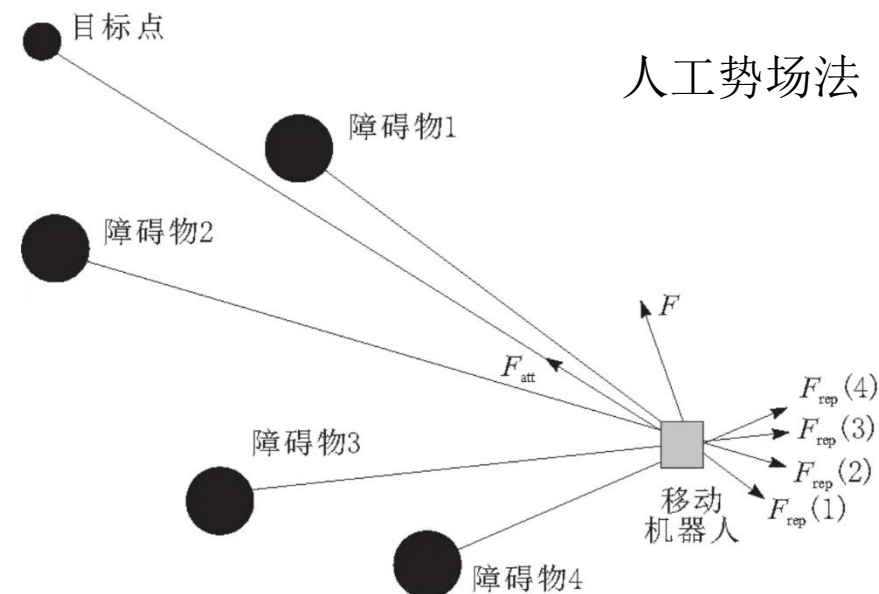
c) 势场力类算法——构造合适的势场，目标引力，障碍斥力，通过合力方向确定机器人方向，合力大小确定机器人加速度。



Dijkstra算法



A\*算法



人工势场法

# 第八讲 2-路径规划算法概述

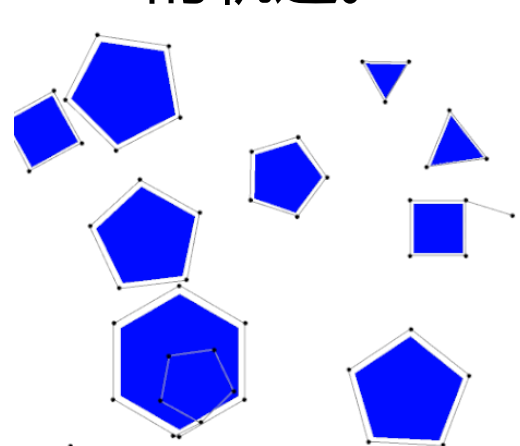
## 2, 空间采样算法 (对空间不需要使用栅格化)

### a) 状态空间采样 (关节角空间/笛卡尔空间) —— **RRT**、PRM

在搜索空间中随机采样，并通过图或树的方式连接这些采样点，去掉不满足约束的连线，最后基于图或树的搜索算法生成最优路径。

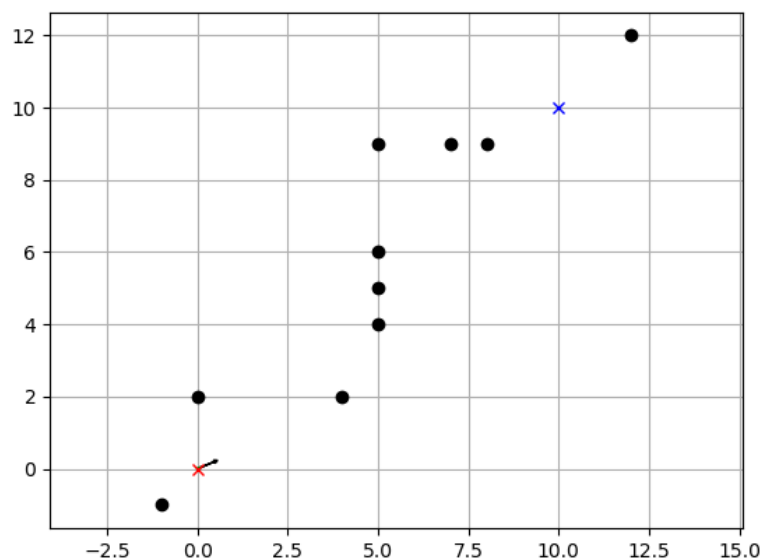
### b) 速度空间采样 (速度空间) —— CVM, DWA

基于运动空间进行采样，并结合运动学模型生成预测轨迹，评估、选择最合适的轨迹。



生成连通图，然后通过图搜索算法找最短路径。

PRM (probabilistic roadmap method)



DWA (Dynamic Window Approach)

先对速度采样，线速度和角速度，然后生成采样轨迹，根据评价函数选最优轨迹  
 (1 **方位角**，机器人在当前的设定的速度下，轨迹末端朝向与目标点之间的角度差距；2, **轨迹末端点与最近障碍物距离**；3, **线速度大小**，等等；)



# 第八讲 3-关节空间与笛卡尔空间描述

**关节空间描述**——采用关节量来描述机器人的运动。

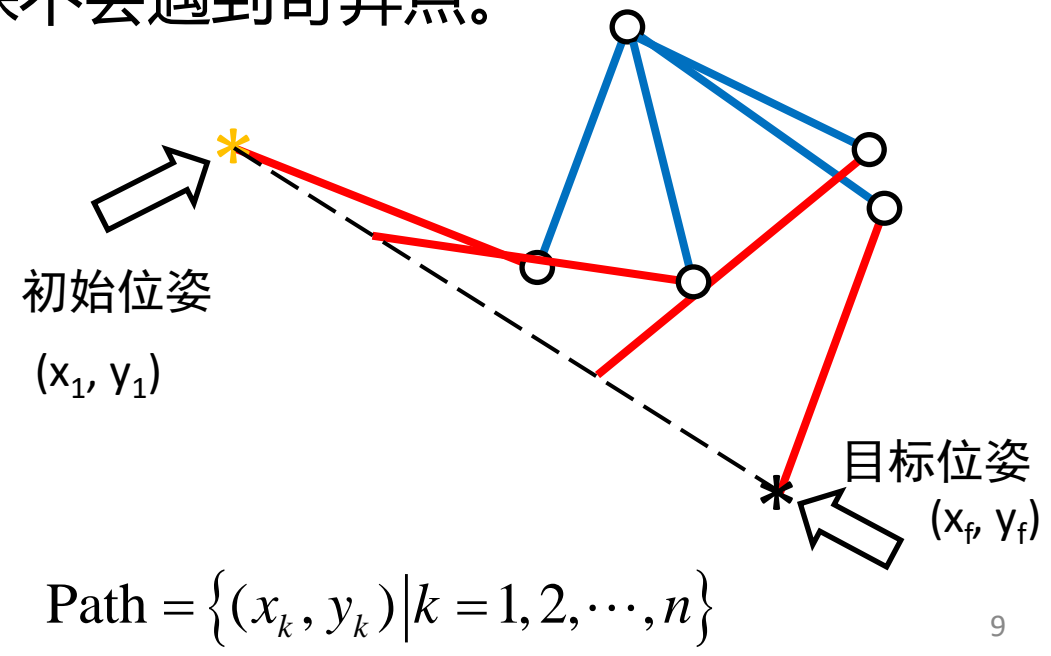
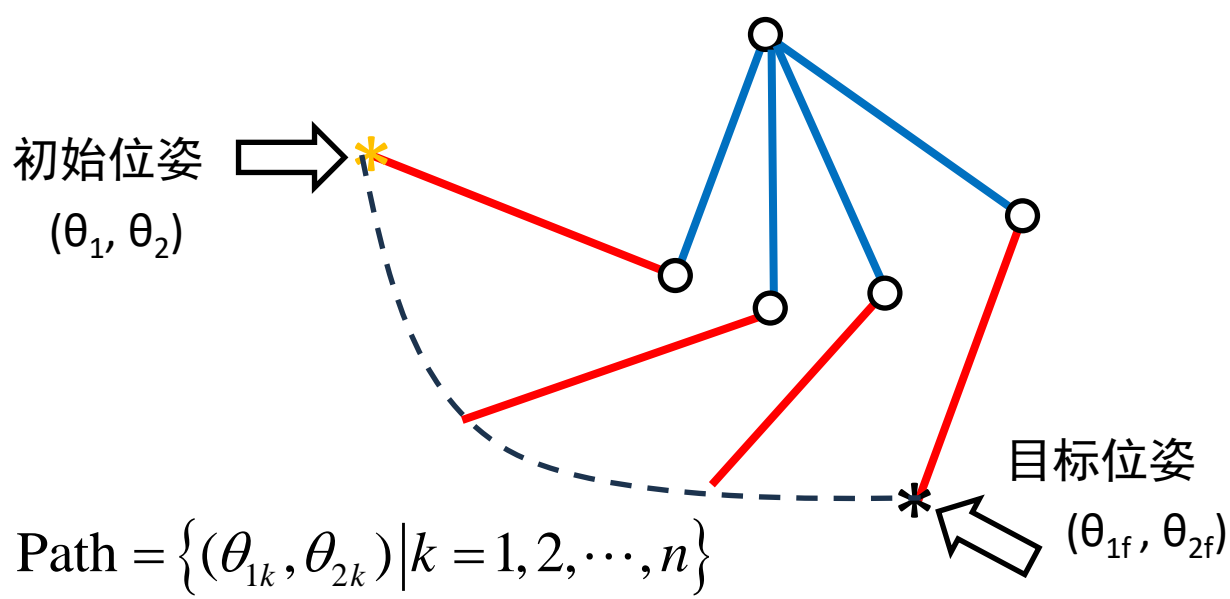
优点：计算简单，快速，不需要用到求解逆运动学。

缺点：机器人末端在初始位置和目标位置之间的运动不可预知。

**笛卡尔空间描述**——采用机器人末端位姿的运动来描述机器人的运动。

优点：具体直观，很容易看到机器人末端执行器的轨迹。

缺点：计算量大，需要用到逆运动学，难以确保不会遇到奇异点。



# 第八讲 3-关节空间与笛卡尔空间描述

## 笛卡尔空间规划的三类典型问题。

问题1，路径存在不可到达的点。

问题2，路径上存在奇异点导致某些关节速度趋于无穷大。

问题3，关节转动极限导致，某些解无法使用，造成机器人关节运动不连续。

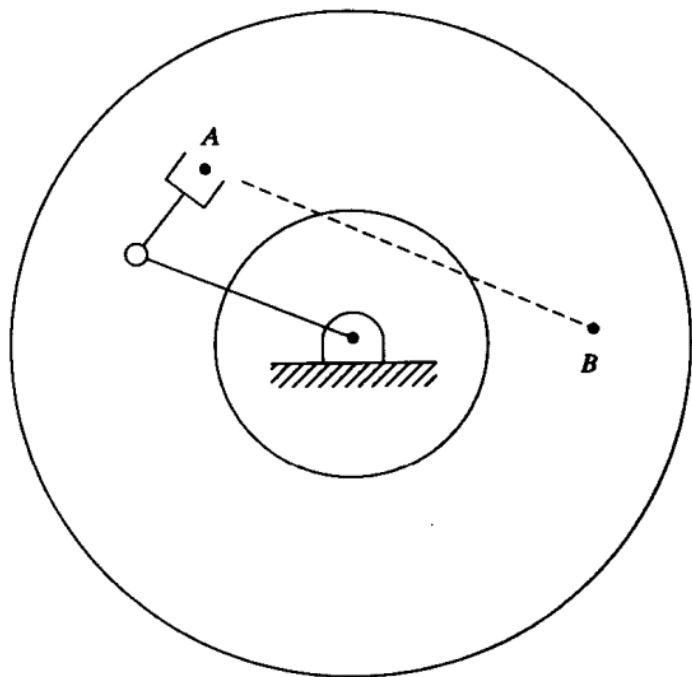


图7-12 笛卡儿路径问题之一

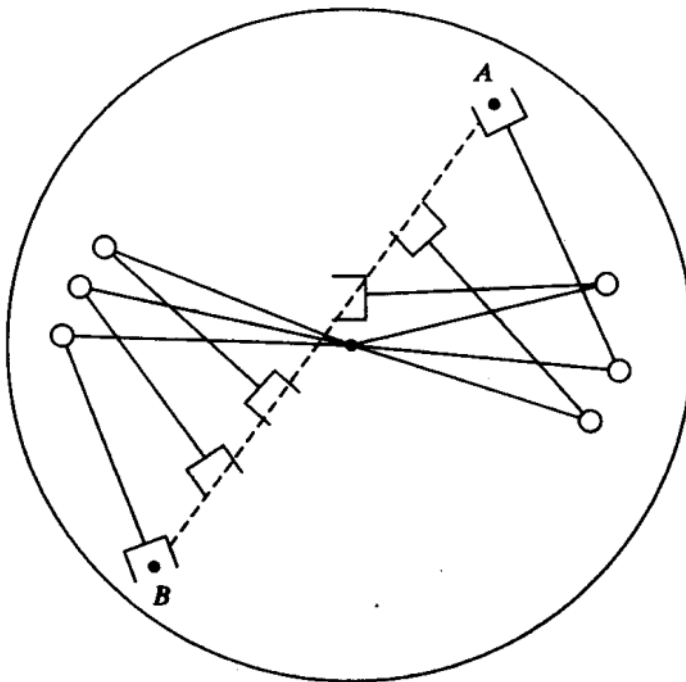


图7-13 笛卡儿路径问题之二

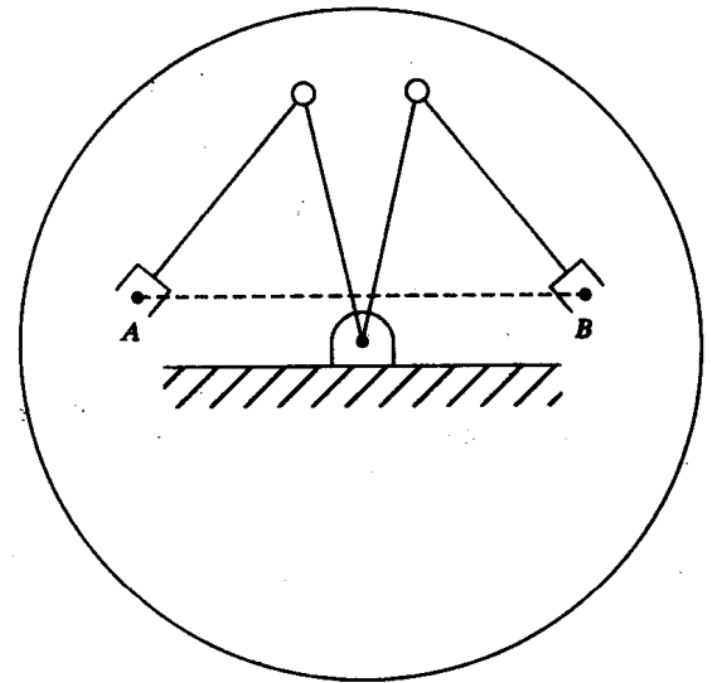


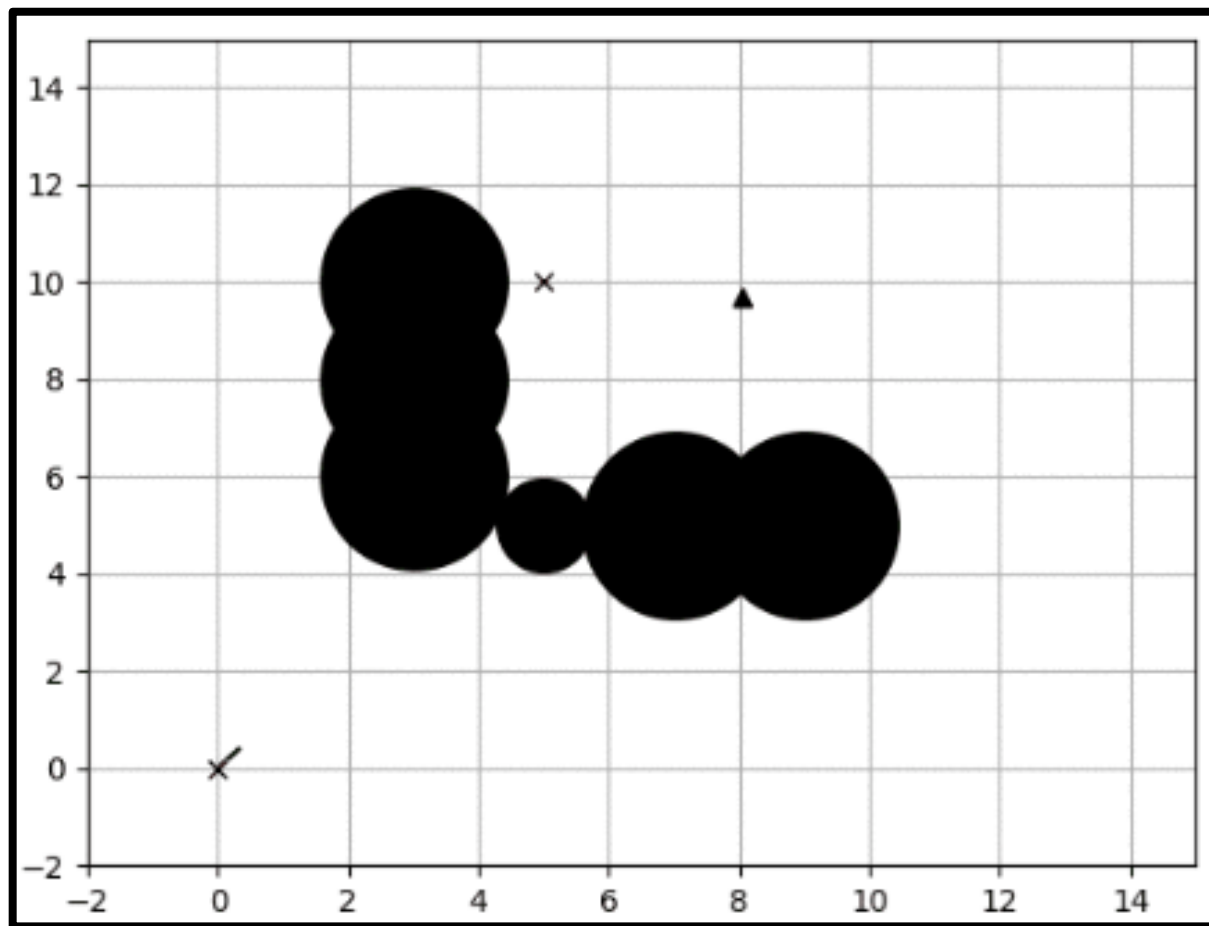
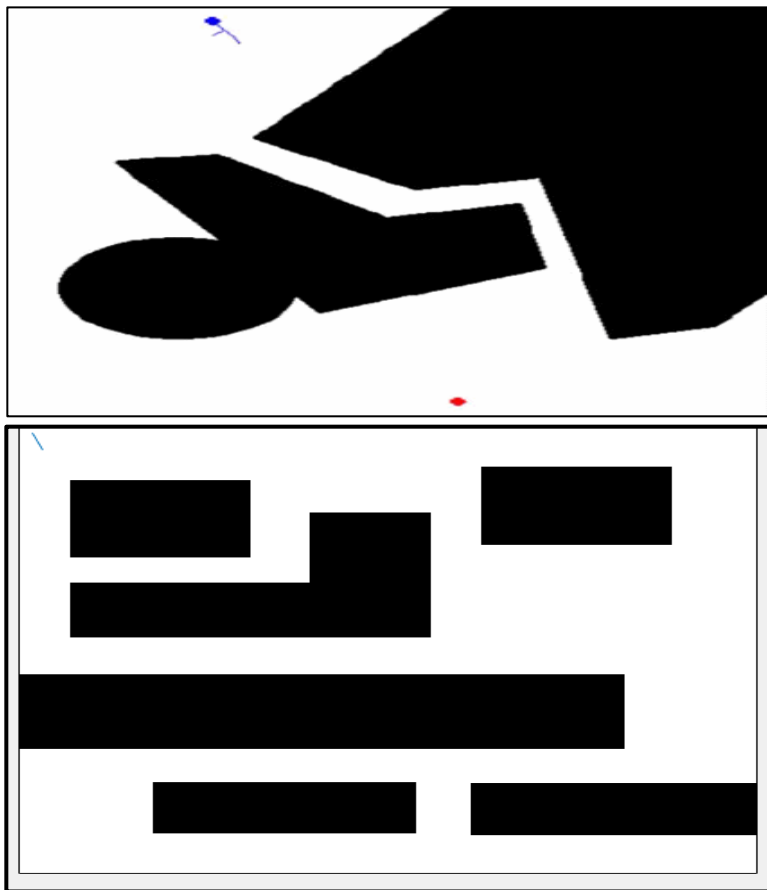
图7-14 笛卡儿路径问题之三

# 第八讲 4-RRT路径规划算法

RRT- (Rapidly Exploring Random Tree) 快速扩展随机树算法。

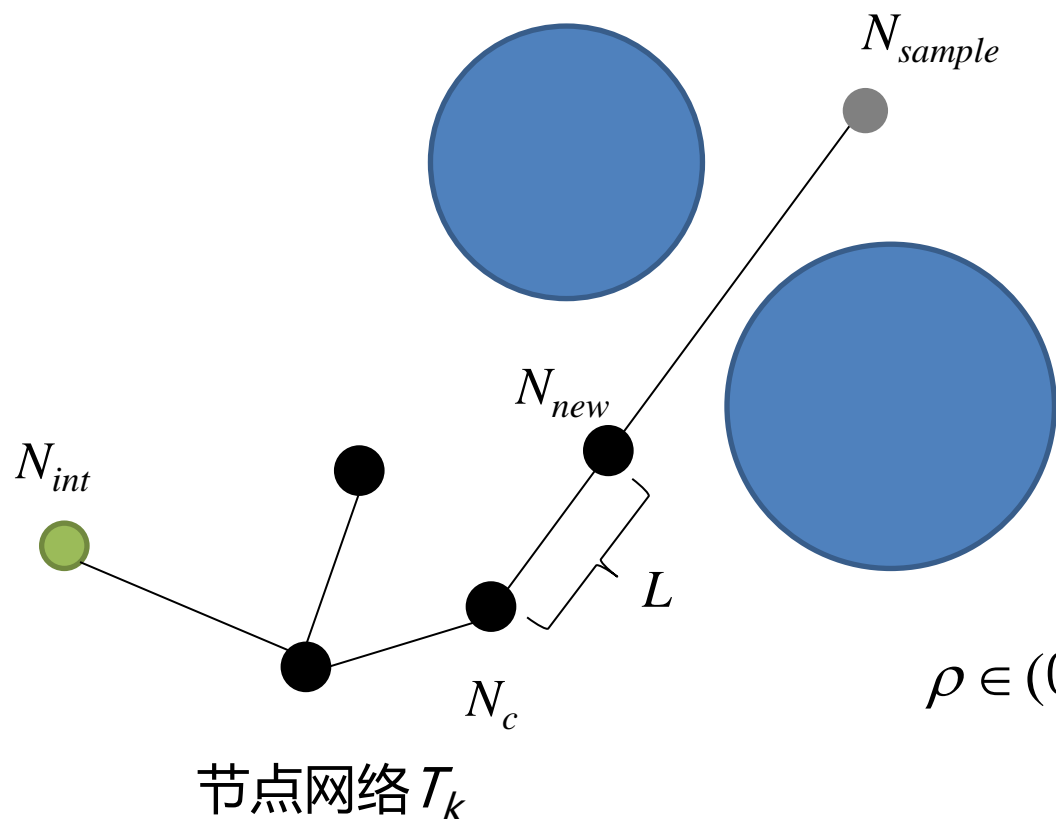
- RRT及改进型RRT是目前机器人领域最常用的路径规划方法。
- 概率完备，但不是最优。

参考代码: <http://rkala.in/codes.php>



# 第八讲 4-RRT路径规划算法

RRT- (Rapidly Exploring Random Tree) 快速扩展随机树算法。

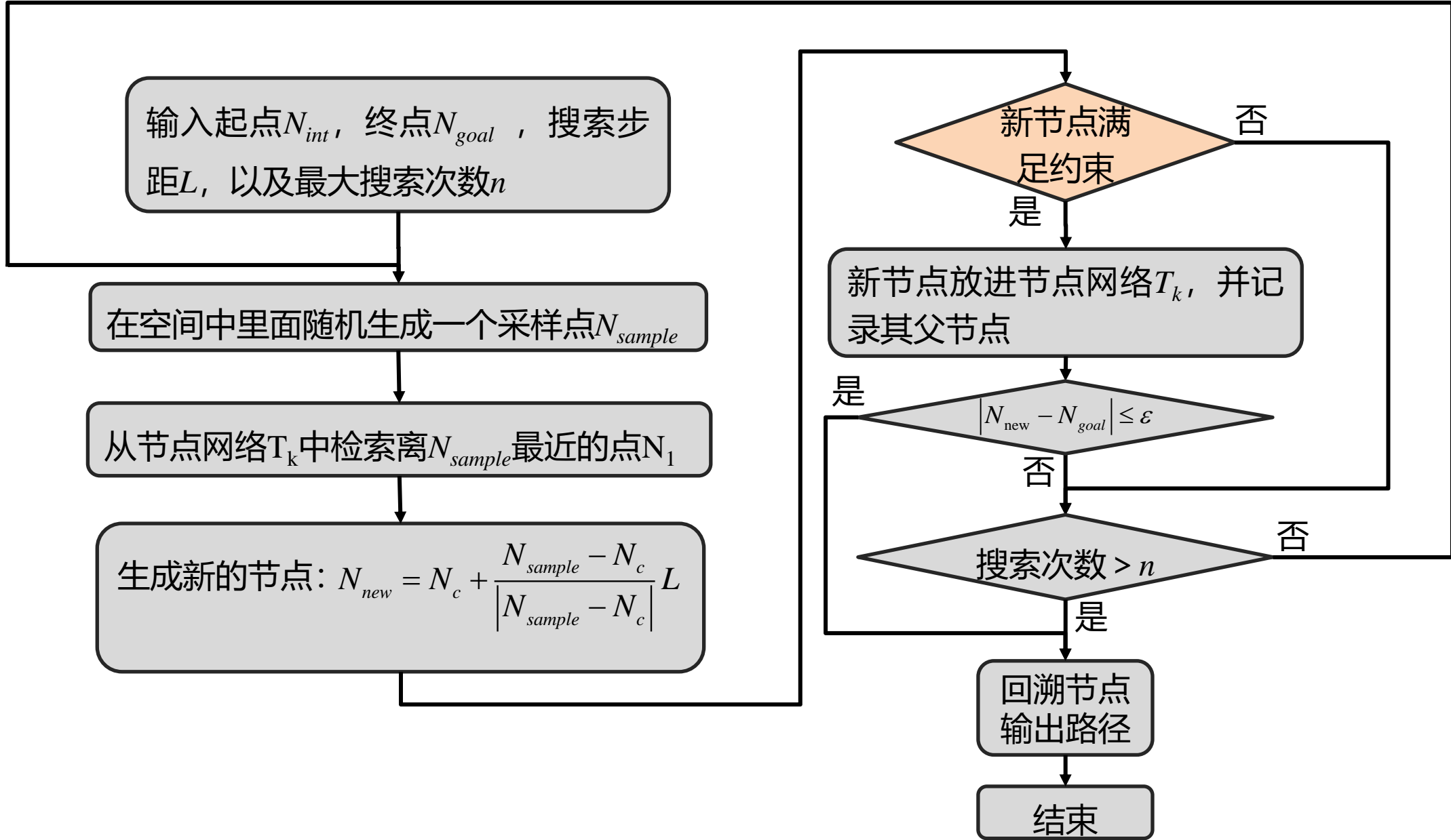


$$N_{sample} = \begin{cases} rand * size(map) & rand > (1 - \rho) \\ N_{goal} & rand < (1 - \rho) \end{cases}$$

$$N_{new} = N_c + \frac{N_{sample} - N_c}{|N_{sample} - N_c|} L$$

$\rho \in (0, 1)$  用于使新节点有一定概率向目标点生长  
 $L$  步长用于给定节点的生长长度，决定路径点间的距离。

# 第八讲 4-RRT路径规划算法





# 第八讲 4-RRT路径规划算法

**关节空间RRT:**

起点:  $N_{\text{int}} = [\theta_1^i, \theta_2^i, \dots, \theta_k^i]'$

终点:  $N_{\text{goal}} = [\theta_1^g, \theta_2^g, \dots, \theta_k^g]'$

采样点:  $N_{\text{sample}} = [\theta_1^s, \theta_2^s, \dots, \theta_k^s]'$

新节点: 
$$N_{\text{new}} = \begin{bmatrix} \theta_1^c \\ \theta_2^c \\ \vdots \\ \theta_k^c \end{bmatrix} + \frac{1}{\sqrt{(\theta_1^s - \theta_1^c)^2 + (\theta_2^s - \theta_2^c)^2 + \dots + (\theta_k^s - \theta_k^c)^2}} \begin{bmatrix} \theta_1^s - \theta_1^c \\ \theta_2^s - \theta_2^c \\ \vdots \\ \theta_k^s - \theta_k^c \end{bmatrix} L$$

**笛卡尔空间RRT:**

起点:  $N_{\text{int}} = [p_i, R_i]'$

终点:  $N_{\text{goal}} = [p_g, R_g]'$

采样点:  $N_{\text{sample}} = [p_s, R_s]'$

$$R_{\text{err}} = R_c^T R_s$$
$$\omega^{\text{err}} = \begin{cases} \begin{bmatrix} 0 & 0 & 0 \end{bmatrix}^T & (R^{\text{err}} = E) \\ \frac{\theta}{2 \sin \theta} \begin{bmatrix} r_{32} - r_{23} \\ r_{13} - r_{31} \\ r_{21} - r_{12} \end{bmatrix} & (R^{\text{err}} \neq E) \end{cases}$$
$$\theta = \arccos(\frac{r_{11} + r_{22} + r_{33} - 1}{2})$$

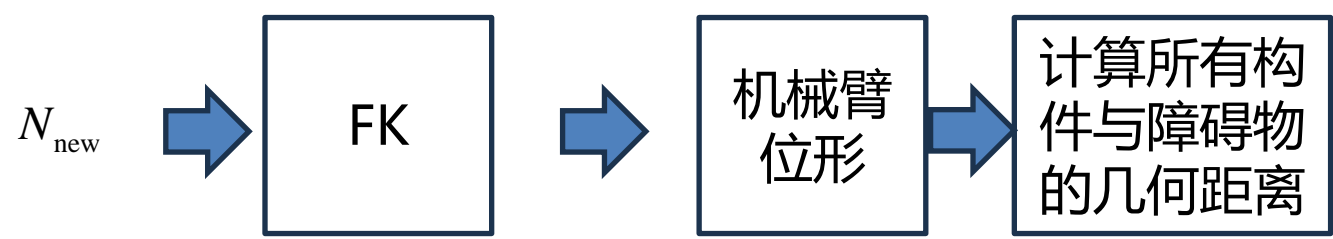
新节点:

$$\Delta N = \begin{bmatrix} \Delta p \\ \Delta \omega \end{bmatrix} = \frac{1}{\|p_c - p_s\|^2 + \|\omega_{\text{err}}\|^2} \begin{bmatrix} p_c - p_s \\ \omega_{\text{err}} \end{bmatrix} L$$
$$\Delta \omega \Rightarrow \Delta R$$
$$N_{\text{new}} = \begin{bmatrix} p_c + \Delta p \\ R_c \Delta R \end{bmatrix}$$

# 第八讲 4-RRT路径规划算法

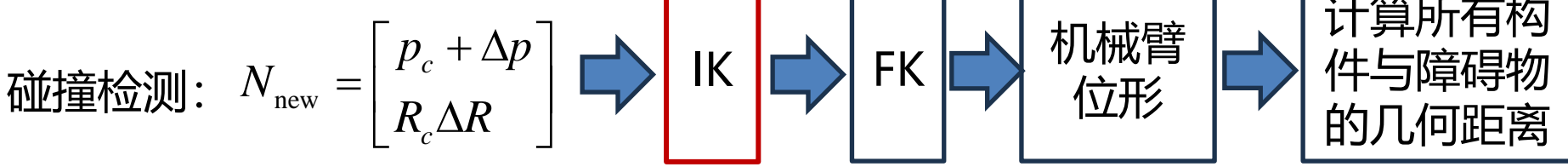
## 关节空间RRT:

碰撞检测:



抵达终点判断:  $|N_{\text{new}} - N_{\text{goal}}| \leq \varepsilon$

## 笛卡尔空间RRT:



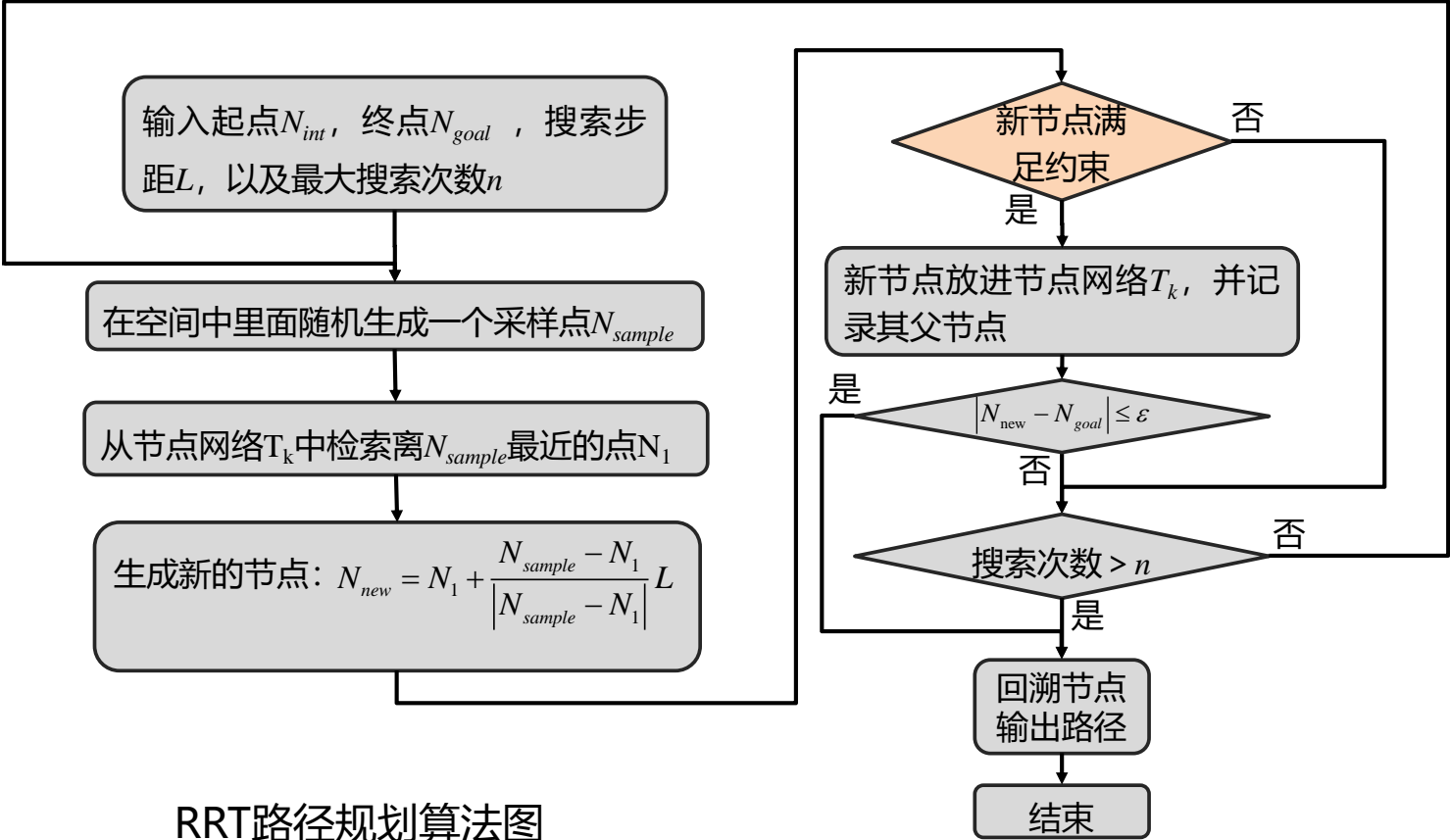
抵达终点判断:  $|N_{\text{new}} - N_{\text{goal}}| \leq \varepsilon$

$$err = \|p^{err}\|^2 + \|\omega^{err}\|^2 \leq \varepsilon$$

$$R_{err} = R_c^T R_s$$
$$\omega^{err} = \begin{cases} \begin{bmatrix} 0 & 0 & 0 \end{bmatrix}^T & (R^{err} = E) \\ \frac{\theta}{2 \sin \theta} \begin{bmatrix} r_{32} - r_{23} \\ r_{13} - r_{31} \\ r_{21} - r_{12} \end{bmatrix} & (R^{err} \neq E) \end{cases}$$
$$\theta = \arccos(\frac{r_{11} + r_{22} + r_{33} - 1}{2})$$

关节RRT更加常用，计算效率更高，因为中间少了计算逆解的步骤。

# 第八讲 4-RRT路径规划算法



当输入的点集N为机械臂末端位置信息[p,R]时为笛卡尔空间（末端）RRT规划。

- 优点：能够直观地规划出末端位置的轨迹，实现末端的避障。
- 缺点：采用末端RRT需要求解逆解才能判断机械臂构型，从而是否与障碍物发生碰撞，**计算量大**。

当输入的点集N为机械臂的关节角序列，例如，对于三自由度的机械臂，输入的N为 $[\theta_1, \theta_2, \theta_3]$ 时，为关节空间RRT规划。

- 优点：能够直接利用关节角根据机械臂的构型判断是否与障碍物发生碰撞，无需求解逆解，**计算方便**。
- 缺点：机器人在初始位置和目标位置之间的末端轨迹不可预知。

附录：

A\*算法动画：[Introduction to the A\\* Algorithm \(redblobgames.com\)](http://redblobgames.com)