



# 第五讲 避障规划

熊蓉

浙江大学 控制科学与工程学院

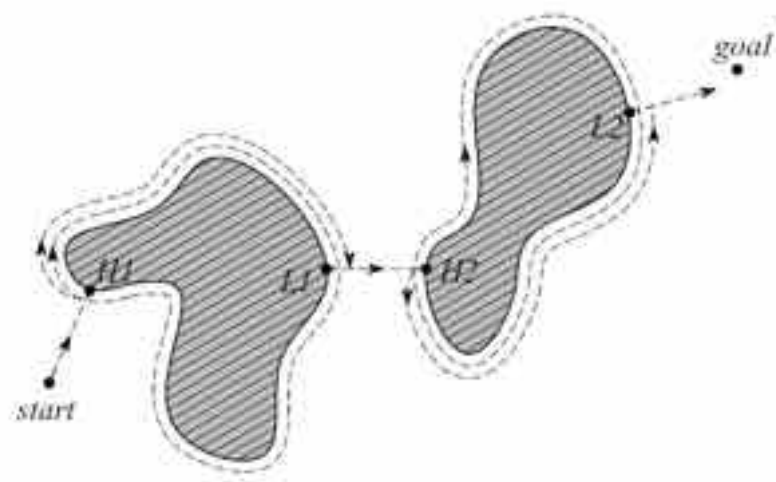
# 避障规划

- 根据所得到的实时传感器测量信息，调整路径/轨迹以避免发生碰撞
- 也称为反应式避障
- 主要方法：
  - Bug算法
  - 向量势直方图法
  - 动态窗口法
  - 人工势场法



# BUG算法

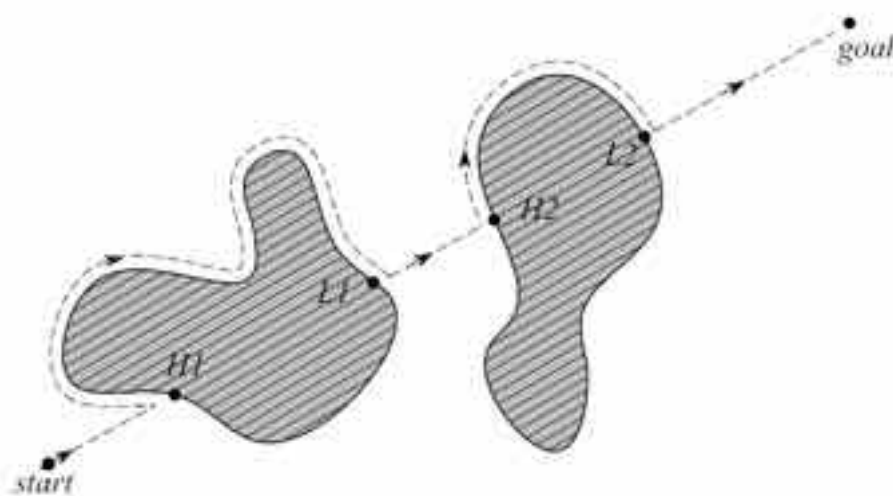
- 基本思想：绕着到目标点路上的每个障碍物的轮廓移动



**Bug1算法：**当遇到障碍物堵塞路径时，机器人绕着障碍物移动一周，找到障碍物上最靠近目标的点，然后再移动到该点，从该点向目标点移动。

优点：非常简单，可以确保机器人能够到达任何可达目标；  
缺点：为了找到障碍物上最靠近目标的点，需要先绕着障碍物移动一周，效率很低。

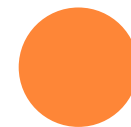
# BUG算法



**Bug2算法：**当出现障碍物时，机器人绕着障碍物移动，一旦机器人可以按照原起始点到目标点路径移向目标时，就立即脱离障碍物

优点：在一定程度上缩短了整个路径

存在问题：机器人每一时刻的行为仅仅依据该时刻的传感器信息，当信息量不够时难以实现高效鲁棒的避障。



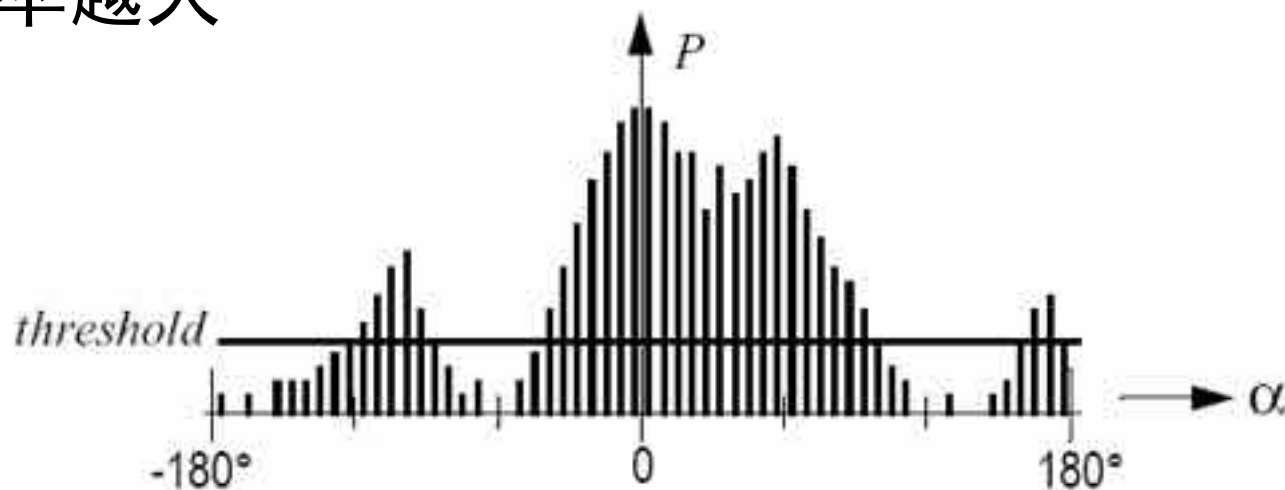
# 向量势直方图法

- 基本思路：融合多帧环境信息寻找最优可行通路

- 基本步骤：

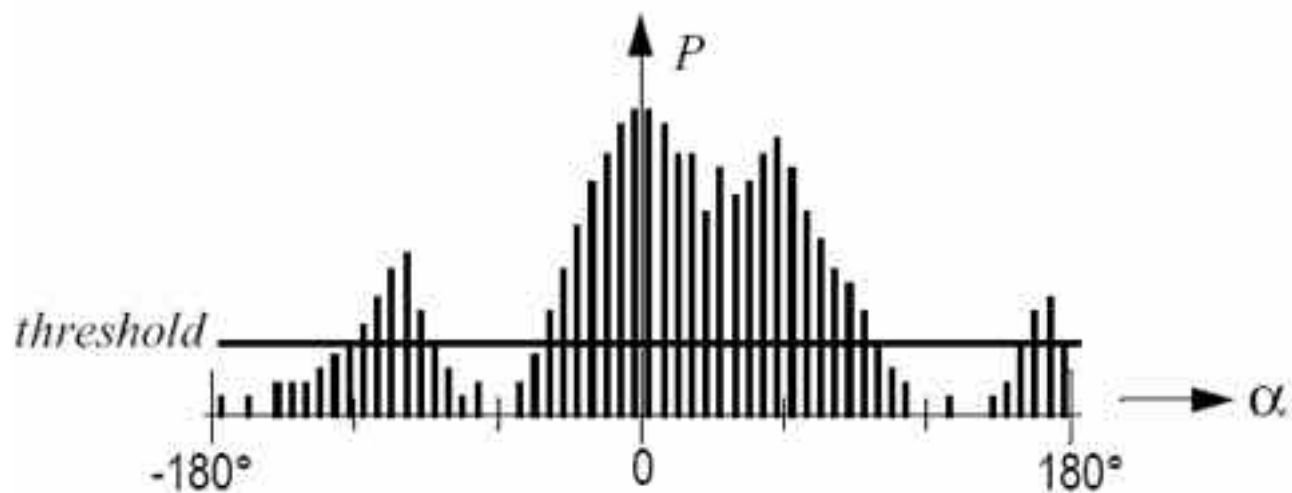
1. 算法内构建并维护机器人周围环境的局部栅格地图

2. 根据占用栅格地图计算障碍物概率直方图，障碍物越近概率越大



## 向量势直方图法

3. 根据直方图，识别所有可以让机器人通过的通道，  
然后对每个通道计算成本，选择具有最低成本的通道，  
得到导航方向



# 向量势直方图法

## ○ 成本计算示例：

$$G = a \cdot target\_direction + b \cdot wheel\_orientation + c \cdot previous\_direction$$

其中考虑了三方面因素

**target\_direction** : 路径与目标之间的对齐量

**wheel\_orientation**: 新方向和当前机器人方向的差异量

**Previous\_direction**: 原来选择方向和新方向之间的差异量

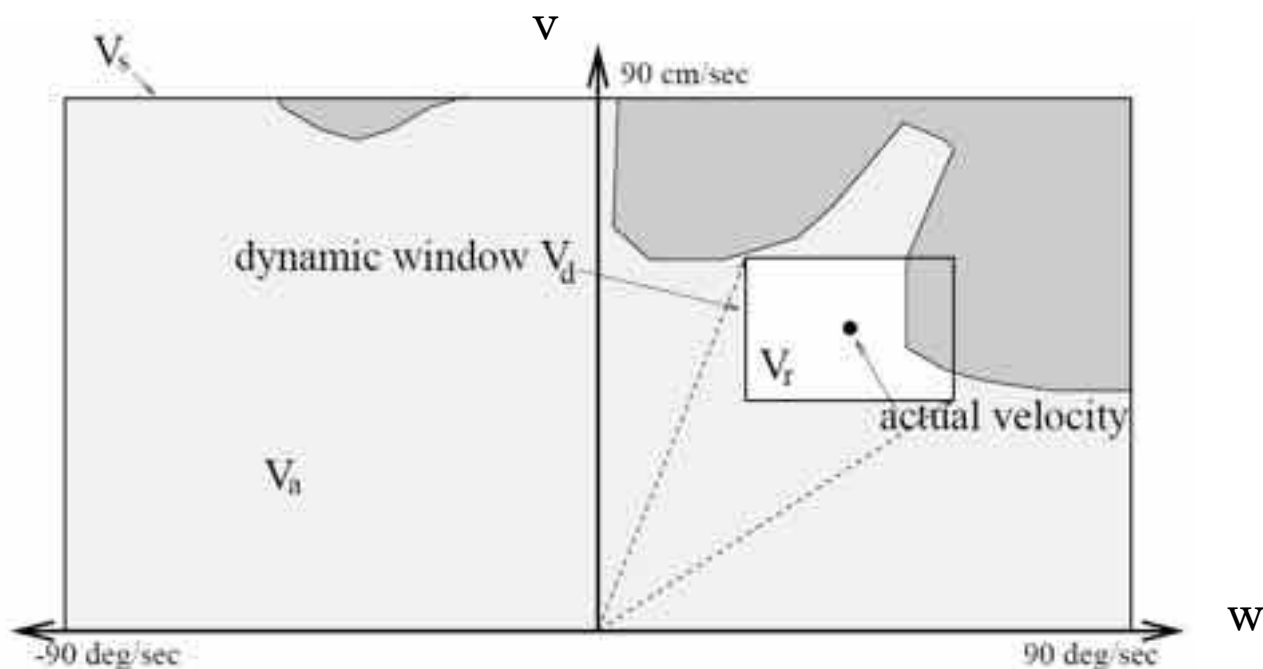
通过a,b,c进行权重调节。

这三个量都是角度量，因此不需要做归一化



# 动态窗口法(DWA, Dynamic Window Algorithm)

- 1997年由波恩大学Thrun课题组提出
- 基本思想：在速度空间中搜索适当的平移速度和旋转速度指令  $(v, w)$

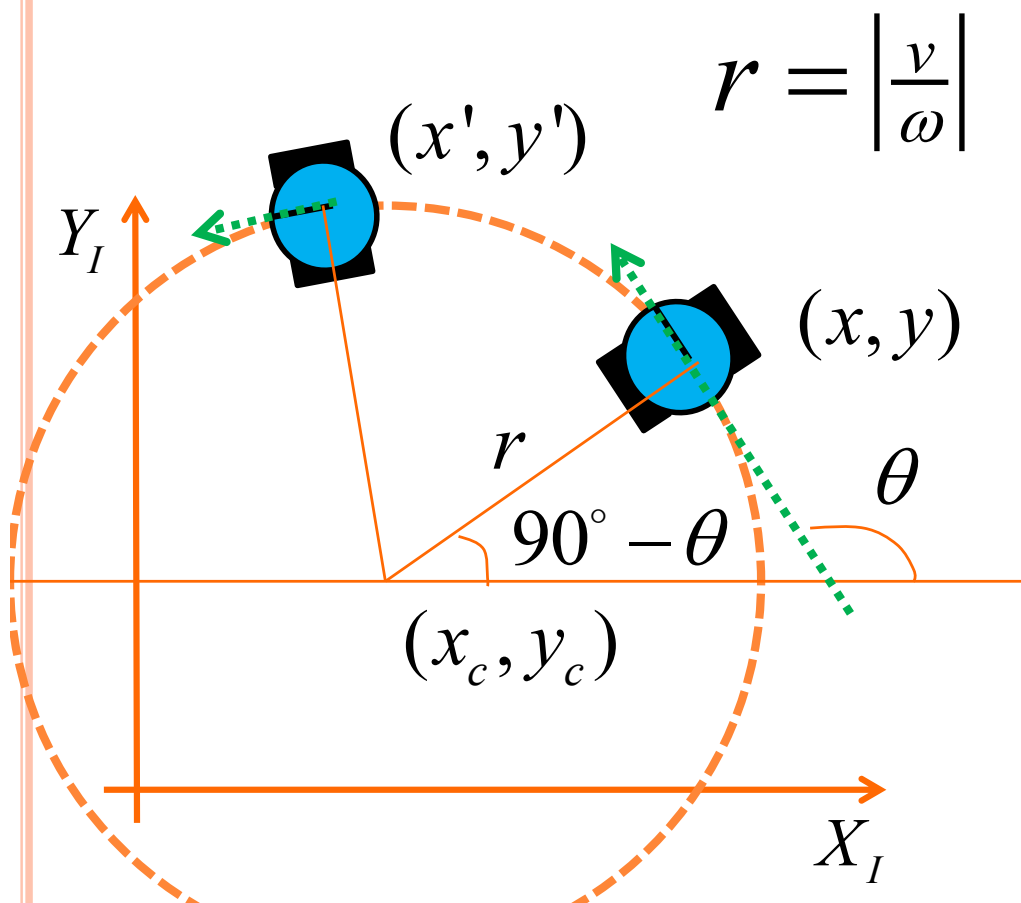




# 机器人的速度控制运动模型

假设没有噪声

设控制时间间隔为  $\Delta t$ ，时间间隔内机器人速度和角速度保持不变，则机器人绕着半径为  $r$  的圆周运动



$$x = x_c + r \sin(\theta)$$

$$y = y_c - r \cos(\theta)$$

$$x' = x_c + r \sin(\theta')$$

$$= x_c + r \sin(\theta + \omega \Delta t)$$

$$y' = y_c - r \cos(\theta')$$

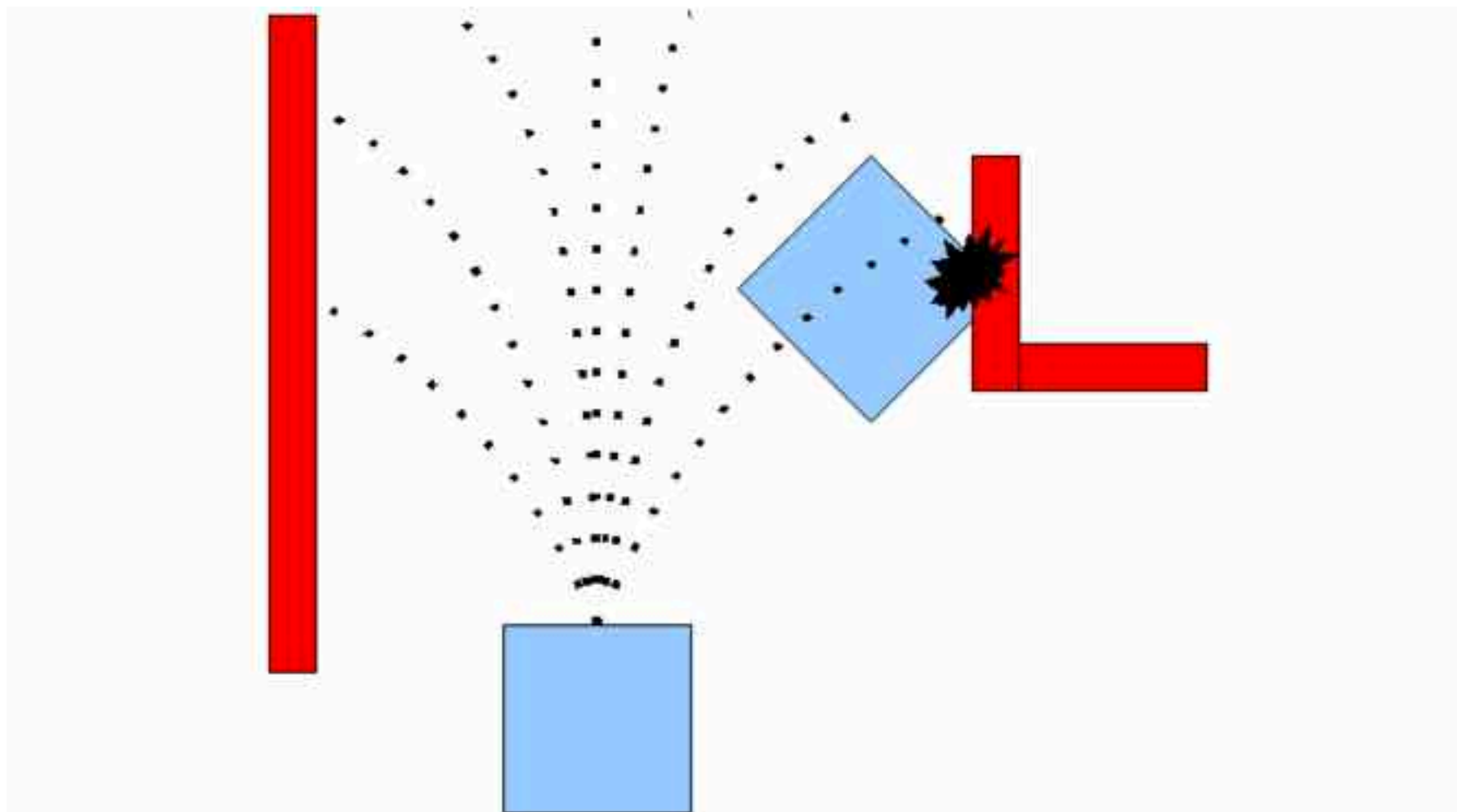
$$= y_c - r \cos(\theta + \omega \Delta t)$$

# 机器人的速度控制运动模型

$$\begin{pmatrix} x' \\ y' \\ \theta' \end{pmatrix} = \begin{pmatrix} x - \frac{v}{\omega} \sin(\theta) + \frac{v}{\omega} \sin(\theta + \omega \Delta t) \\ y + \frac{v}{\omega} \cos(\theta) - \frac{v}{\omega} \cos(\theta + \omega \Delta t) \\ \theta + \omega \Delta t \end{pmatrix}$$

无噪声下的理想运动模型

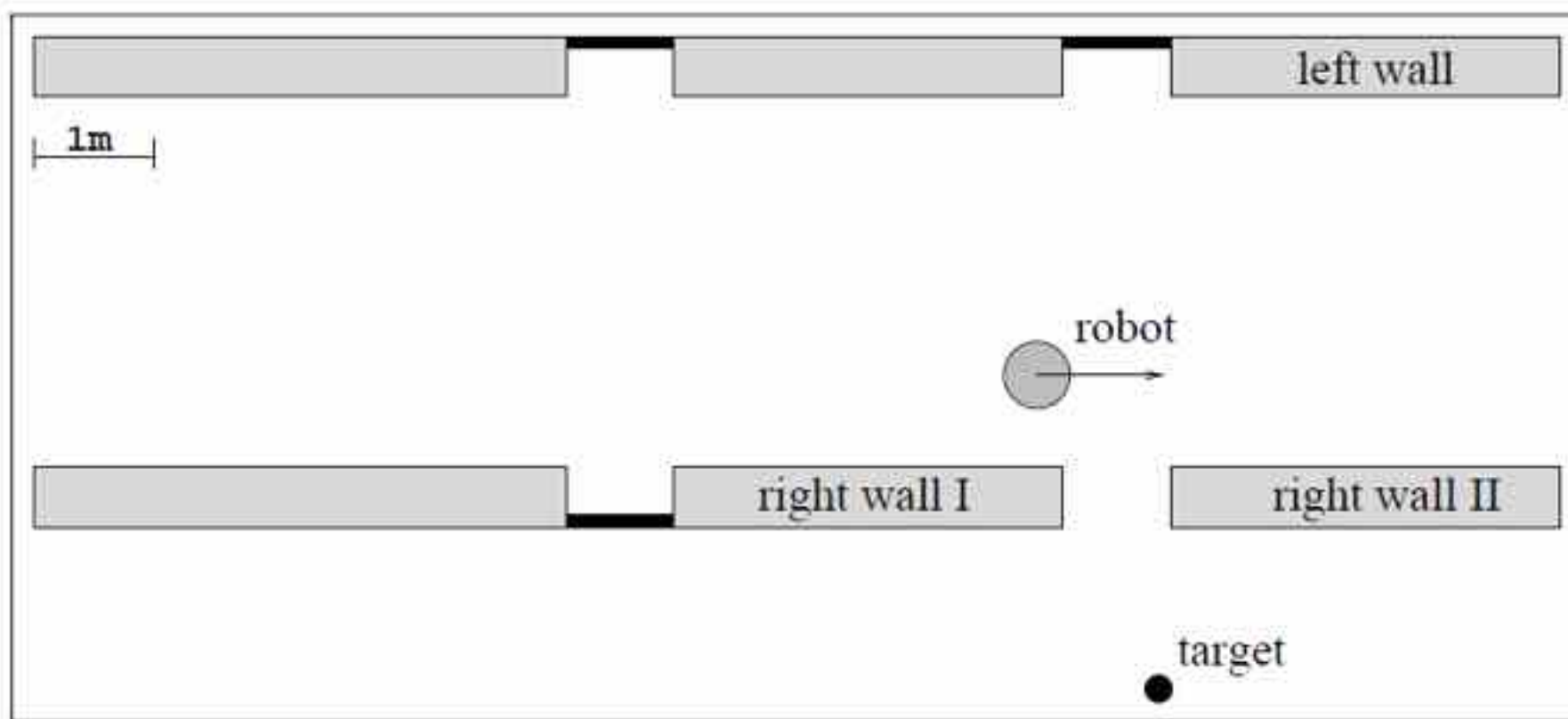




不同的速度指令  $(v, w)$  会得到不同的运动半径，同样的时间间隔到达不同的终止位置。有些位置是安全的，有些会与障碍物发生碰撞

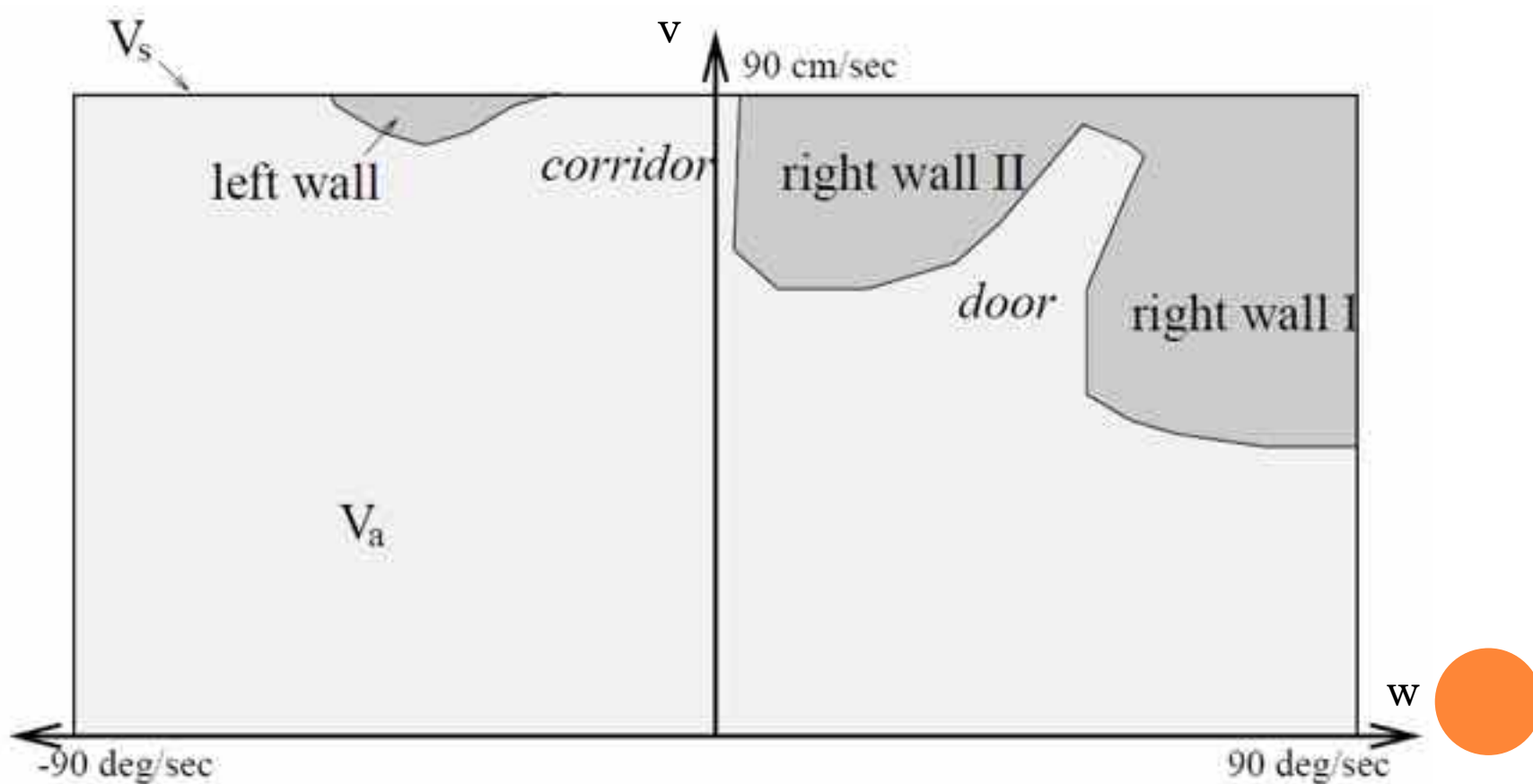
# 实现步骤

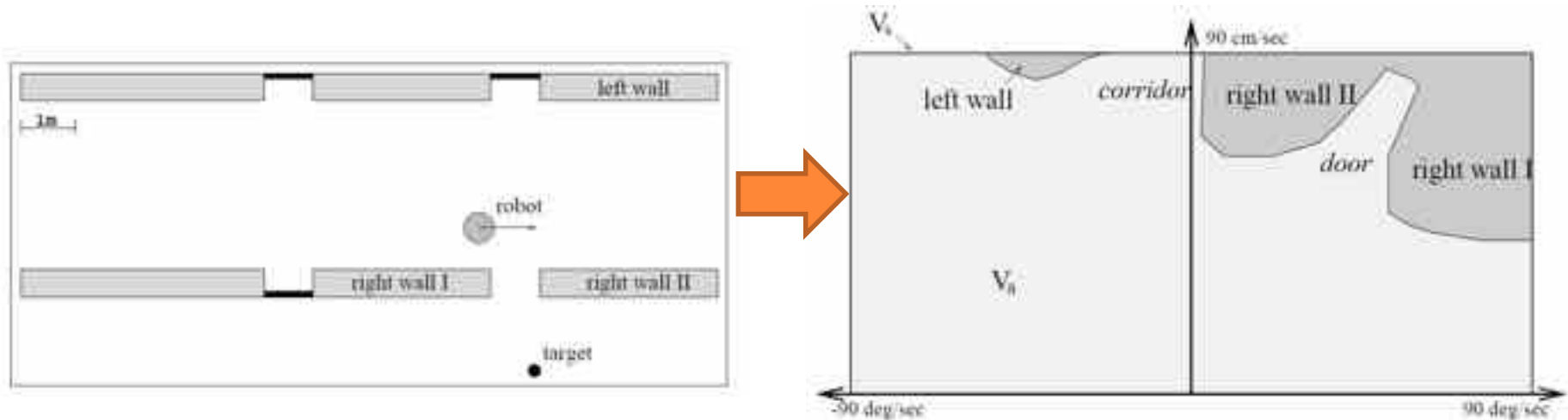
## (1) 基于速度控制运动模型，构建可行的速度空间



# 实现步骤

## (1) 基于速度控制运动模型，构建可行的速度空间





$$V_a = \left\{ (v, \omega) \mid v \leq \sqrt{2 \cdot \text{dist}(v, \omega) \cdot \dot{v}_b} \wedge \omega \leq \sqrt{2 \cdot \text{dist}(v, \omega) \cdot \dot{\omega}_b} \right\}$$

可以让机器人停止不与障碍物相碰的可行速度集合

$\dot{v}_b$  最大平移加速度

$\dot{\omega}_b$  最大旋转加速度

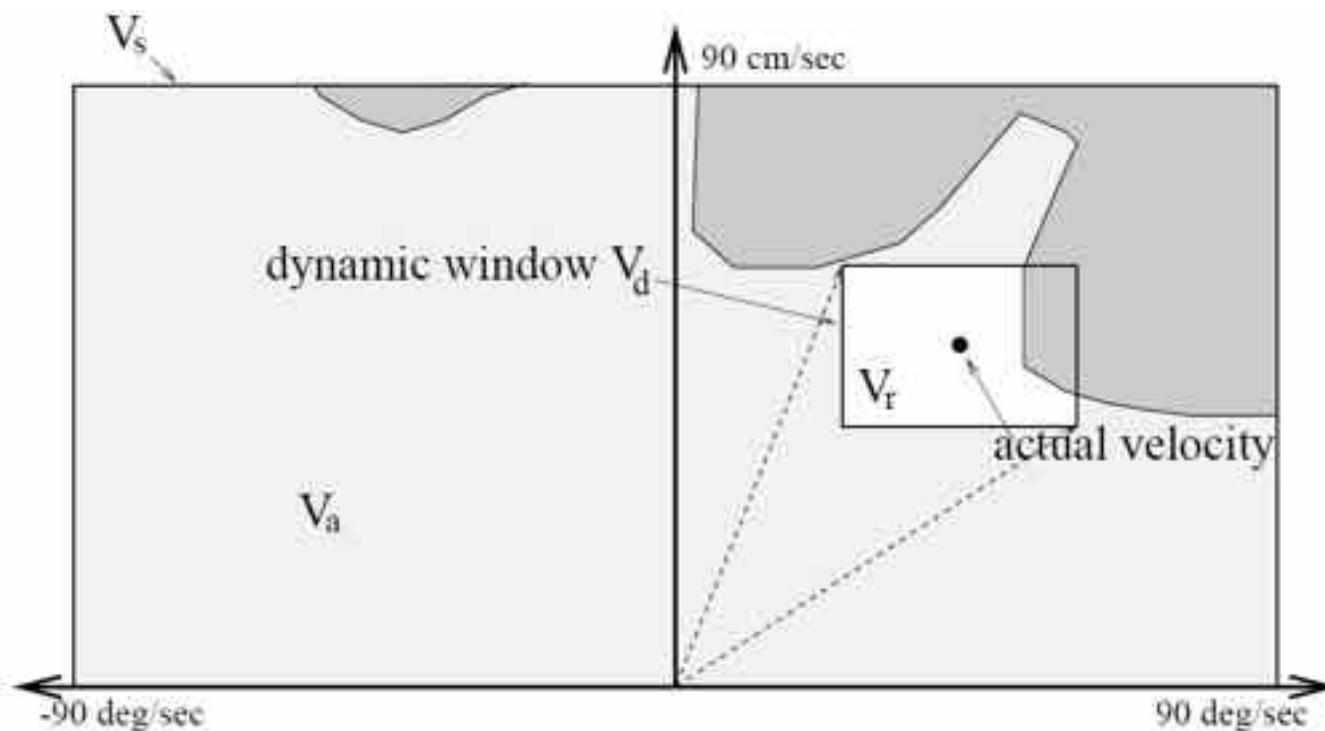
$\text{dist}(v, \omega)$  表示速度配置  $(v, \omega)$

所对应圆弧上最近障碍物的距离



# 实现步骤

- (2) 考虑到机器人在运动过程中最大加速度的约束，在当前速度配置处以固定的小时间间隔开一个速度窗口空间



# 实现步骤

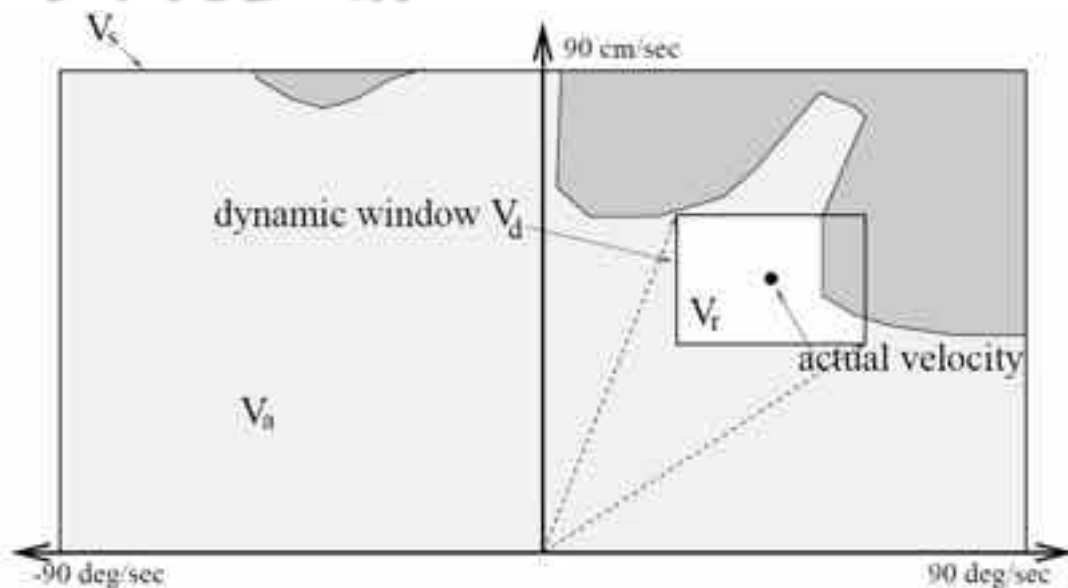
(2) 考虑到机器人在运动过程中最大加速度的约束, 在当前速度配置处以固定的小时间间隔开一个速度窗口空间

$$V_d = \{(v, \omega) \mid v \in [v_l, v_h] \wedge \omega \in [\omega_l, \omega_h]\}$$
$$\begin{cases} v_l = v_a - a_{v\max} \times \Delta t \\ v_h = v_a + a_{v\max} \times \Delta t \\ \omega_l = \omega_a - a_{\omega\max} \times \Delta t \\ \omega_h = \omega_a + a_{\omega\max} \times \Delta t \end{cases}$$





# 实现步骤



(3)结合机器人速度约束，获得可行速度空间为

$$V_r = V_a \cap V_d \cap V_s$$

$$V_a = \left\{ (v, \omega) \mid v \leq \sqrt{2 \cdot \text{dist}(v, \omega) \cdot \dot{v}_b} \wedge \omega \leq \sqrt{2 \cdot \text{dist}(v, \omega) \cdot \dot{\omega}_b} \right\}$$

$$V_d = \{(v, \omega) \mid v \in [v_l, v_h] \wedge \omega \in [\omega_l, \omega_h]\}$$

$$V_s = \{(v, \omega) \mid v \in [-v_{\max}, v_{\max}] \wedge \omega \in [-\omega_{\max}, \omega_{\max}]\}$$



# 实现步骤

## (4) 在可行速度空间中选择最优的速度控制指令

$$evaluation(v, \omega) = \alpha \cdot heading(v, \omega) + \beta \cdot dist(v, \omega) + \gamma \cdot velocity(v, \omega)$$

$$\alpha + \beta + \gamma = 1 (\alpha \geq 0, \beta \geq 0, \gamma \geq 0)$$

$heading(v, \omega)$  朝向目标点 : 保证机器人朝目标点运动

$dist(v, \omega)$  远离障碍物 : 保证机器人避开障碍物, 安全不碰撞

$velocity(v, \omega)$  速度最大化 : 保证机器人以最大速度运动

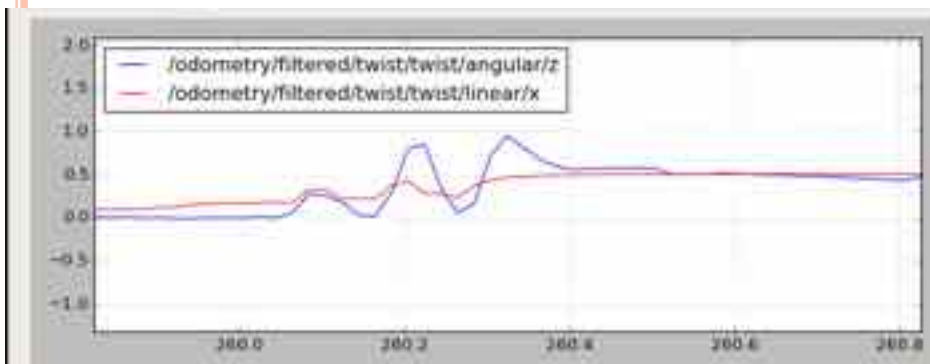
注意各项的归一化处理



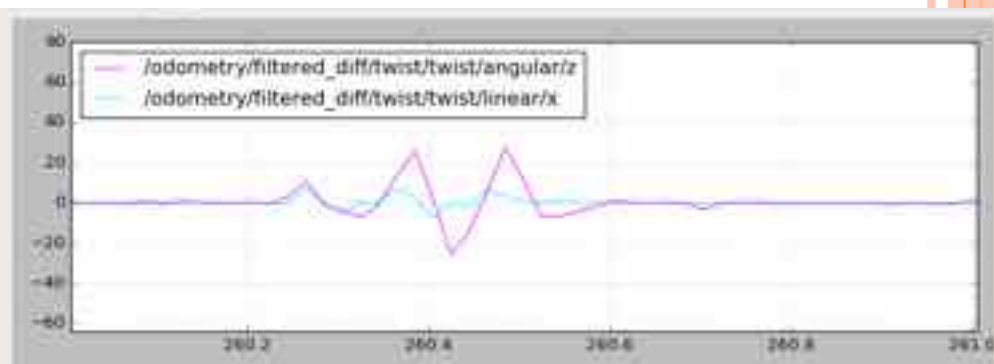
# 动态窗口法

- 存在问题：

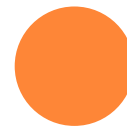
因为它是根据单步信息数据来计算期望速度，容易导致加速度和jerk值（加加速度）非常大，导致机器人无法实现或者运动不平稳，称为运动不舒适

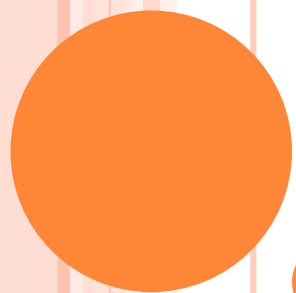


速度



加速度





**END !**