DWA 路径规划

1 基础理论

1.1 DWA 算法

1.1.1 机器人运动模型

在动态窗口法中，要模拟机器人的轨迹，需要知道机器人的运动模型，动态窗口法采用的是假设两轮移动机器人不是全向移动的，其轨迹是一段一段的圆弧或者直线（旋转速度为 0），一对（v, w）就代表一个圆弧轨迹，轨迹方程为（当不等 w 于 0）：

$$\begin{cases} x_{t+1} = x_t - \frac{v}{w}\sin(\theta_t) + \frac{v}{w}\sin(\theta_t + w\Delta t) \\ y_{t+1} = y_t + \frac{v}{w}\cos(\theta_t) - \frac{v}{w}\cos(\theta_t + w\Delta t) \\ \theta_{t+1} = \theta_t + w\Delta t \end{cases} \tag{1}$$

当 w 等于 0 时：

$$\begin{cases} x_{t+1} = x_t + v\Delta t \\ y_{t+1} = y_t + v\Delta t \end{cases} \tag{2}$$

1.1.2 动态窗口的形成

速度和角速度采样是动态窗口法的核心，对速度和角速度进行均匀采样，形成 (v, w) 的几何，根据机器人本身硬件、环境和运动状态条件的限制，可以将采样的数量控制在一定范围内,我们让 (v_{odom}, w_{odom}) 表示机器人本时刻的速度和角速度，$(\dot{v}, \dot{w})$ 表示机器人的最大线加速度和最大角加速度。

(1) 机器人硬件限制所限制的搜索空间为：

$$V_s = \{(v, w) \mid v \in (v_{\min}, v_{\max}), w \in (w_{\min}, w_{\max})\} \tag{3}$$

(2) 运动状态限制,由于受电机力矩影响，存在最大加减速度，因此在一个模拟周期内，存在一个搜索空间：

$$V_d = \{(v, w) \mid v \in (v_{odom} - \dot{v}\Delta t, v_{odom} + \dot{v}\Delta t), w \in (w_{odom} - \dot{w}\Delta t, w_{odom} + \dot{w}\Delta t)\} \tag{4}$$

(3) 环境限制，由于考虑到机器人的安全问题，即机器人必须能够在 (v, w) 形成的预测轨迹与障碍物点云之间的最小距离内停下：

$$V_a = \{(v, w) \mid v <= \sqrt{2dist(v, w)\dot{v}} \;\&\&\; w <= \sqrt{2dist(v, w)\dot{w}}\} \tag{5}$$

那么总的搜索空间为：

$$V = V_a \cap V_b \cap V_s \tag{6}$$

1.1.3 评价函数：

为了选取动态窗集合内最优的一组的 (v, w)，设置一个评价函数，对所有组合进行评价，

评价函数分为四项，即方位角评价函数，速度评价函数，安装评价函数和目标距离评价函数。

（1）方位角评价函数($head(v,w)$ ：即机器人执行当前 (v,w) 生成的预测轨迹末端与目标点之间的连线与轨迹末端机器人朝向之间的夹角差，如图 1 所示：

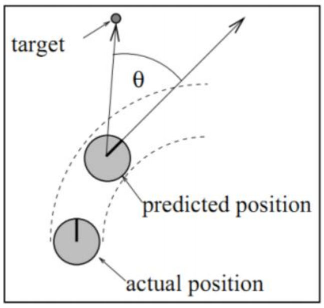

图 1 方位角评价示意图

$head(v,w)$ 函数的表达式为：

$$head(v,w) = \arctan(1,|\theta|) \tag{7}$$

角度差值越小，则 $heading(v,w)$ 的值越大，机器人会越朝向目标点运动。

（2）安全评价函数 $Odist(v,w)$ ：用来评价生成的预测轨迹到障碍物的最小距离。它的表达式为：

$$Odist(v,w) = \min(distToObstacle(v,w)) \tag{8}$$

$distToObstacle(v,w)$ 表示轨迹上的点到障碍物的距离列表，求取其中的最小值，最小值越大，安全评价函数就越大，说明预测轨迹到障碍物距离就越大。安全性就越高。

(3)速度评价函数 $speed(v,w)$ ：用于评价当前选择的速度大小。它的表达式为：

$$speed(v,w) = v / v_{\max} \tag{9}$$

速度越大，速度评价函数就越高。

(4) 目标距离评价函数 $Gdist(v,w)$ ：用来评价生成的预测轨迹到目标点之间的最小距离，它的表达式为：

$$Gdist(v,w) = \frac{1}{\min(distToGoal(v,w))} \tag{10}$$

其中 $distToGoal(v,w)$ 为轨迹上的点到目标点之间的距离列表，求其中的最小值，最小值约小，目标距离评价函数越大，说明轨迹距目标点越近。

(4)归一化处理：得到的三个评价函数是不可以直接相加的，需要归一化平滑处理，平滑方程为：

$$\begin{cases} \mathrm{smooth(head(i))} = \dfrac{\mathrm{head(i)}}{\sum_{i=1}^{n} head(i)} \\ \mathrm{smooth(Odist(i))} = \dfrac{\mathrm{Odist(i)}}{\sum_{i=1}^{n} Odist(i)} \\ \mathrm{smooth(speed(i))} = \dfrac{\mathrm{speed(i)}}{\sum_{i=1}^{n} speed(i)} \\ \mathrm{smooth(Gdist(i))} = \dfrac{\mathrm{Gdist(i)}}{\sum_{i=1}^{n} Gdist(i)} \end{cases} \tag{9}$$

其中 n 为生成轨迹总和，i 为第 i 条轨迹。

（5）总评价函数的表达式为：

$$G(v,w) = headW * head(v,w)_{som} + OdistW * Odist(v,w)_{som} + speedW * speed(v,w)_{som} + GdistW * Gdist(v,w)$$

其中，$headW, OdistW, speedW, GdistW$ 为权重系数。通过对搜索空间内的 (v,w) 进行评价，选取总评价函数值最大的一组作为控制命令发给机器人。

2 参数说明

vehicleParam

maxSpeed: 机器人最大行驶速度

minSpeed：机器人最小行驶速度

maxAngularSpeed：机器人最大角速度

maxAcc ： 机器人最大行驶加速度

maxAngularAcc： 机器人最大角加速度

dwaParam：

speedReso ： 动态窗中线速度搜索的分辨率

angularReso ：动态窗中角速度搜索的分辨率

Δt ：采样时间

predictTime ：轨迹时间

headW： $head(v,w)$ 函数在总评价函数中的权重系数，即权重系数 headW 越大，$head(v,w)$ 的值在总评价函数中的占比就越大，dwa 算法就会更倾向于选择预测轨迹末端朝向与目标点夹角更小的 (v,w) 组合。

OdistW： $Odist(v,w)$ 函数在总评价函数中的权重系数，即权重系数 OdistW 越大，

$Odist(v,w)$ 的值在总评价函数中的占比就越大，dwa 算法就会更倾向于选择预测轨迹到最近障碍物距离更大的 (v,w) 组合。

speedW：$speed(v,w)$ 函数在总评价函数中的权重系数，即权重系数 speedW 越大，$speed(v,w)$ 的值在总评价函数中的占比就越大，dwa 算法就会更倾向于选择速度更大的 (v,w) 组合。

GdistW：$Gdist(v,w)$ 函数在总评价函数中的权重系数，即权重系数 GdistW 越大，$Gdist(v,w)$ 的值在总评价函数中的占比就越大，dwa 算法就会更倾向于选择预测轨迹到目标点距离更小的 (v,w) 组合。

safeDistance：预测轨迹到最近障碍物的距离的阈值，当实际的最小 distance 小于 safeDistance，此 (v,w) 的组合将会被排除，不会考虑。

3 轨迹预测公式

1. 第一种：切线模型

第一种预测轨迹的方式为直线法，即将预测轨迹离散的分成一段一段的直线，也是现在比较常用的轨迹预测公式为如下所示：

$$\begin{cases} x_{t+1} = x_t + v \bullet \Delta t \\ y_{t+1} = y_t + v \bullet \Delta t \\ \theta_{t+1} = \theta_t + w \bullet \Delta t \end{cases}$$

这种是比较常用的一种轨迹预测公式模型，模型要求机器人不是全向的机器人，精度不高

2. 第二种：圆弧模型(使用中)

为 DWA 基础理论推出中的轨迹预测模型，它将轨迹预测称一段一段圆弧，圆弧的半径为 $r = v / w$，模型要求依然不是全向机器人，相比于第一种模型，更加精确，具体公式如下(公式 1)：

$$\begin{cases} x_{t+1} = x_t - \frac{v}{w}\sin(\theta_t) + \frac{v}{w}\sin(\theta_t + w\Delta t) \\ y_{t+1} = y_t + \frac{v}{w}\cos(\theta_t) - \frac{v}{w}\cos(\theta_t + w\Delta t) \\ \theta_{t+1} = \theta_t + w\Delta t \end{cases}$$

3. 第三种：割线模型

它假设机器人先转一半角度，即 $\frac{w}{2}$，然后沿着此方向运动 Δt，最后再转一半角度。那么我们按照此思路，可以直接写出切线模型的方程，模型要求依然不是全向机器人：

$$\begin{cases} x_{t+1} = x_t + v \bullet \cos(\theta_t + w/2) \bullet \Delta t \\ y_{t+1} = y_{\mathrm{t}} + v \bullet \sin(\theta_t + w/2) \bullet \Delta t \\ \theta_{t+1} = \theta_t + w \bullet \Delta t \end{cases}$$