

自己动手写全套无人驾驶算法系列（二）机器人底盘

一、概述

1.1 机器人底盘概述

目前无人车机器人底盘有很多种，比如差分驱动，阿克曼，四轮驱动，三轮驱动等，实际过程中用的比较多的是差分驱动和阿克曼转向，差分适用于扫地机器人，迎宾机器人等，阿克曼转向适用于无人汽车。本人同样在 github 实现了上述两种运动模型，可见：

https://github.com/wujiazheng2020/WJZ_Chassis

如果有问题可以联系我：wujiazheng2020@163.com

QQ 群：710805413

二、差分驱动模型

2.1 程序概述

在 github 那个程序里，需要机器人用 USB 或者通过在机器人上放一个 MCU 如 stm32，通过 stm32 读取驱动器/编码器的信息传输给 PC 段/MPU 端，而 PC 端收到后发送 odom 里程计消息和相应 tf(坐标转换)，同时如果 PC 端收到上层算法的消息，则发送给机器人的 MCU 端，该程序是 PC 端程序，因为 stm32 程序很简单就不写了。这种模型你只能控制左右轮子的速度，其之和/2 为 v ，之差/2r 为 w ， r 为旋转半径，，测量底盘可得，为两轮子之间的距离的一半。

2.2 模型推导

其实这个模型在概率机器人第 5 节中有详细的推导，我的第一章

https://blog.csdn.net/qq_38588806/article/details/88174642 也对其有所介绍。

下面推导 motion_model_velocity 算法和 sample_motion_model_velocity 算法。
像往常一样，对数学细节不感兴趣的读者可跳过本节从 5.4 节继续学习。推导从机器运动产生的模型开始，然后导出采样和对任意 x_t 、 u_t 和 x_{t-1} 计算 $p(x_t | u_t, x_{t-1})$ 的公式。

精确的运动

在转向概率情况前，先从一种理想的无噪声的机器人运动学开始研究。令 $u_t = (v \omega)^T$ 表示时刻 t 的控制。如果两种速度在整个时间间隔 $[t-1, t]$ 都是固定的值，则机器人以如下半径的圆运动：

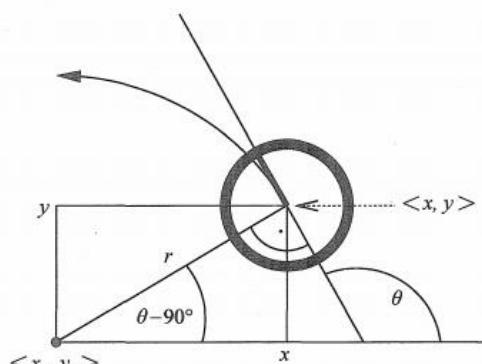


图 5.5 从 $(x \ y \ \theta)^T$ 开始并以恒定速度 v 和 ω 运动的无噪声机器人完成的运动

$$r = \left| \frac{v}{\omega} \right| \quad (5.5)$$

由 $v = wr$ 我们知道给定 v 和 w 小车运动轨迹是一个圆，圆心服从上述公式。 x, y, θ 的描述见上图，那么我们就可以推导得：

圆的中心位于

$$x_c = x - \frac{v}{\omega} \sin \theta$$

$$y_c = y + \frac{v}{\omega} \cos \theta$$

故我们可以得到如下公式：

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

当 w 很小的时候($\rightarrow 0$ 的时候)，可由等价无穷小公式推导得：

```
x = v * cos(theta) * dt
y = v * sin(theta) * dt
theta = w * dt
```

这就是差分驱动运动模型。

二、阿克曼转向模型

2.1 阿克曼转向模型概述

需要无人车支持 CAN 并通过 USBCAN 连接到 PC 端，然后通过其他程序转述，github 这个只是例子，为了展现其运动学模型。

2.2 阿克曼转向模型推导

参考清华的汽车理论 170 页或者北理的模型预测控制。这里就不多加叙述了，最后推导结果如下：(忽视轮差)

```
x = v * cos(theta) * dt
y = v * sin(theta) * dt
```

$\theta = v \times \tan(\text{steer}) / \text{wheel_base}$

其中 wheel_base 为前后轮子中心距离，而我们在 pc 端只能控制车辆的方向盘转角 steer 还有加速度 a 。