# Pure pursuit algorithm

## 1 纯追踪算法

### 学习背景

这个学习记录主要想学习一下关于路径跟随的一些基本控制算法，并进行仿真实验。从最简单的纯追踪算法开始学习。本文的内容主要来源于网络参考，参考网址为：

### 纯追踪

#### 非完整约束运动学模型

本文研究基于两轮差速机器人的路径跟随问题，两轮差速机器人属于非完整约束机器人。此处的非完整约束是指：机器人无法进行横向移动。一般这种受约束的机器人运动参数可以由两个量进行表示：线速度*v*和角速度*w*，在任何一个运动周期中，机器人都以固定的线速度和角速度进行圆周运动。*w* = 0，机器人进行直线运动。

既然是做圆周运动，那么必然存在运动半径，那么可以从这个运动半径下手，对机器人进行运动控制研究。

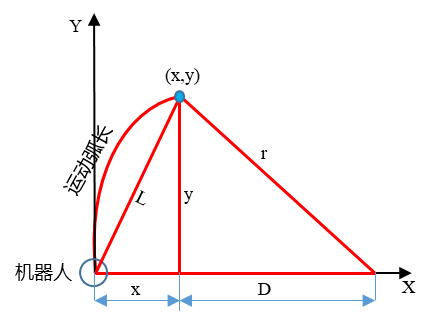


图1-1 非完整机器人运动模型

根据图1-1所示，首先坐标系是基于机器人本体坐标系所建。其中坐标(x, y)也是机器人坐标系下的坐标。现在求取旋转半径。





根据公式(1-1)和(1-2)可得*r* = *L*2/*2x*。

#### 基于pure pursuit的机器人轨迹跟踪

纯追踪算法是受非完整约束机器人做圆周运动的特性所开发出来的运动控制方法。

核心思想为：为机器人提供一个前视搜索距离（或者搜索半径），根据前视距离搜索路径中距离前视距离最近的点。假定机器人的坐标为[*x*0, *y*0, *θ*0]T，其中*θ*0为机器人与全局坐标系的夹角。假定搜索到的最近路径坐标点为[*x*k, *y*k]T，我们的目标是要求取机器人坐标系下，机器人和参考点之间的横向距离*x*，进而根据*r* = *L*2/*2x*求取机器人旋转半径。



图1-2 求取机器人坐标系下的横向距离

下面为计算横向距离*x*的过程公式。







明显的，横向距离公式中*x*的正负代表了曲率的方向（机器人往那边转弯）。根据公式*r* = *L*2/*2x*可得机器人旋转半径*r*。

#### 1.2.3 纯追踪在两轮差速机器人路径跟随的应用

两轮差速机器人运动模型如下：







其中*vr*和*vl*分别为左右两轮的线速度，*d*为机器人两轮间距。根据*r* = *L*2/*2x*。和式可得机器人左右两轮的速度为：





其中公式和公式在实际中运动中需要用到。仿真用不到。

#### 1.2.3前视距离的动态控制

常见的方式与运动速度相关,其中C可为为车辆最小转弯半径，需要根据实际车辆大小进行调参，C多大，车辆运行会平顺，但是与规划路径的偏差会大。过小，车辆容易抖动。



## Matlab仿真实验

首先说一下Matlab做仿真的一个基本流程。

1：产生相应的参考轨迹

2：寻找轨迹上距离当前位置最近的点。

这里要注意，在寻找距离最近路径点的过程中，要注意一些问题，首先在搜索最近点的过程中，没有必要对整条路径进行搜索，只需要对“前方”固定数量点的路径进行搜索即可。

3：根据当前点距离机器人的横向距离（机器人坐标系下），来计算机器人旋转半径，进而计算机器人运行角速度。