简介

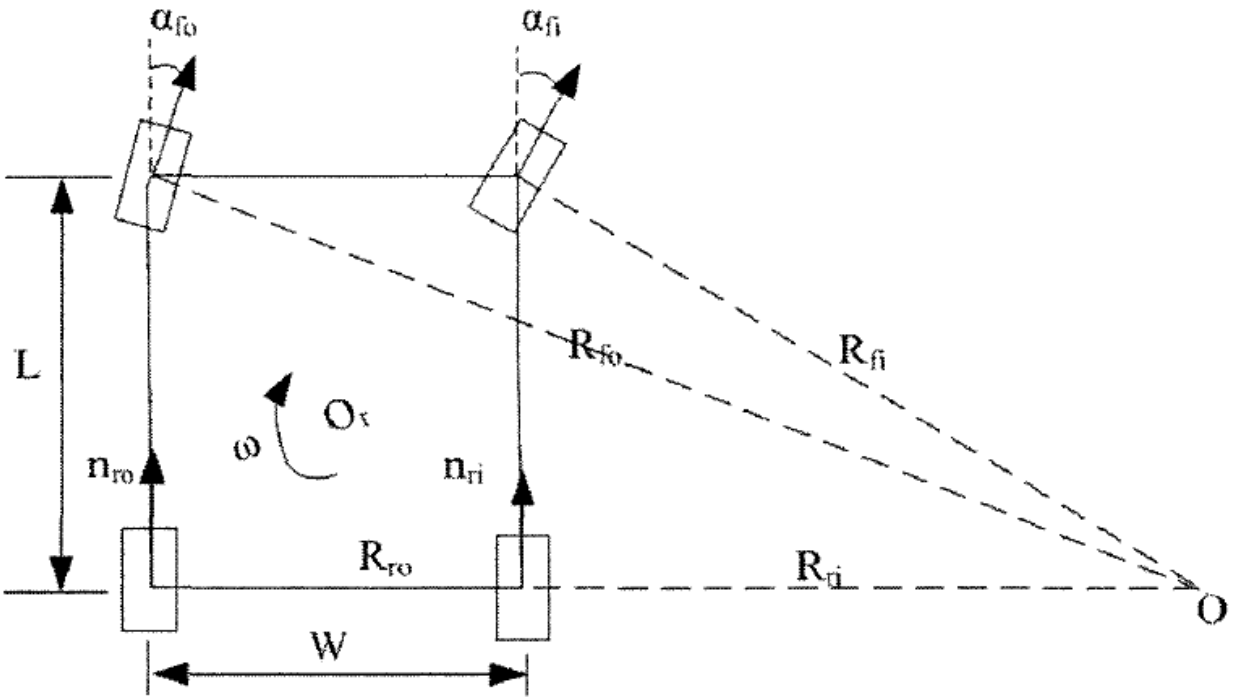
本文档里程推算的模型来自于南网机器人项目中的第二版小车。

为了能够在不用GPS和激光雷达的情况下，且基于初始的位置和航向角信息，确定此时的定位信息，并推算出航迹，因此需要用到里程推算。本项目的最终目标是尽量不使用GPS，紧靠激光雷达融合IMU，里程推算是一个关键的技术节点。

里程推算的输入序列为[x,y,heading,Odometry[4],theta[2],gyroscope]，输出序列为[x,y,heading]。

模型

运动学分析



通过底层反馈回来的车前轮偏转角，求出车后轮的转弯半径Rrj。通过车子所走的弧长，求出绕运动原点的旋转角度theta，然后求出通过以下公式求出旋转之后的坐标:

$$x' = x * cos(theta) - y * sin(theta)$$

$$y' = x * sin(theta) + y * cos(theta)$$

然后计算出当前坐标系下的位移：

$$x = x - x'$$

$$y = y - y'$$

最后将位移投影到世界坐标系下：

$$X+ = cos(Theta) * x - sin(Theta) * y$$

$$y+ = cos(Theta) * y + sin(Theta) * x$$

$$Theta+ = theta$$

伪代码

```
Input: Odometry[4],theta[2],x,y,heading,angle,delta_time
  filtering the Odometry data in [-50,50]
  trace.x = x
  trace.y = y
  trace.thea = heading
  if theta < 0:
      R_left_back = L / tan(theta)
      for i = 0 to 3
          thea += (Odometry[i] - Last_Odometry[i]) / R_left_back
      thea = thea * Scale * 0.5
      x_ = (R_left_back + 0.5 * W) * (cos(thea) - 1) - 0.5 * L * sin(thea)
      y_ = (R_left_back + 0.5 * W) * sin(thea) + 0.5 * L * (cos(thea) - 1)
  else if thea > 0:
      R_left_back = L / tan(theta)
      for i = 0 to 3
          thea += (Odometry[i] - Last_Odometry[i]) / R_left_back
      thea = thea * Scale * 0.5
      x_ = -(R_left_back + 0.5 * W) * (cos(thea) - 1) - 0.5 * L * sin(thea)
      y_ = -(R_left_back + 0.5 * W) * sin(thea) + 0.5 * L * (cos(thea) - 1)
  else:
      for i=0 to 3:
          distance += Odometry[i] - Last_Odometry[i]
      y_ = 0.25 * distance * Scale
  trace.x += cos(trace.thea) * x_ - sin(trace.thea) * y_
  trace.y += cos(trace.thea) * y_ - sin(trace.thea) * x_
  trace.thea += 0.02 *trace.thea + 0.98 * angle
  Last_Odometry = Odometry
Output: trace.x,trace.y,trace.thea
```