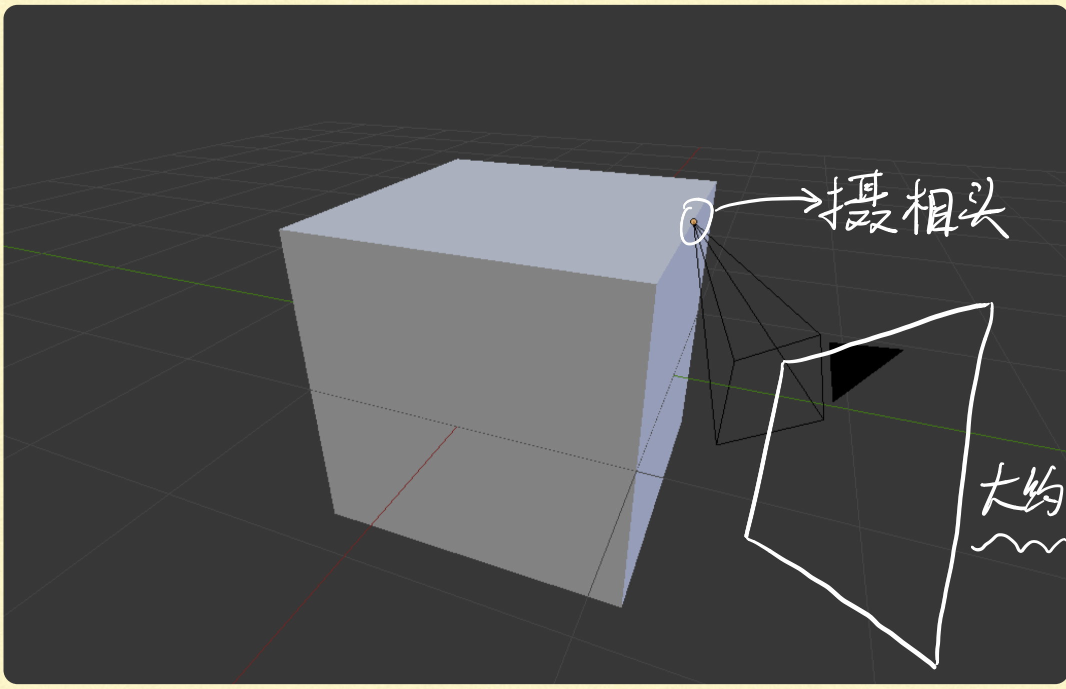


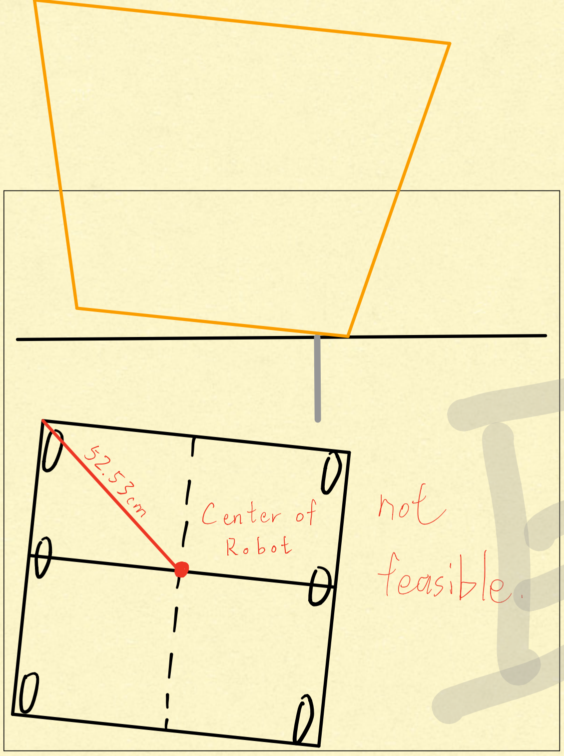
(图片来源:Blender)
方盒代表机器人
摄相头处于远离
机械臂的一侧。

情况分析：

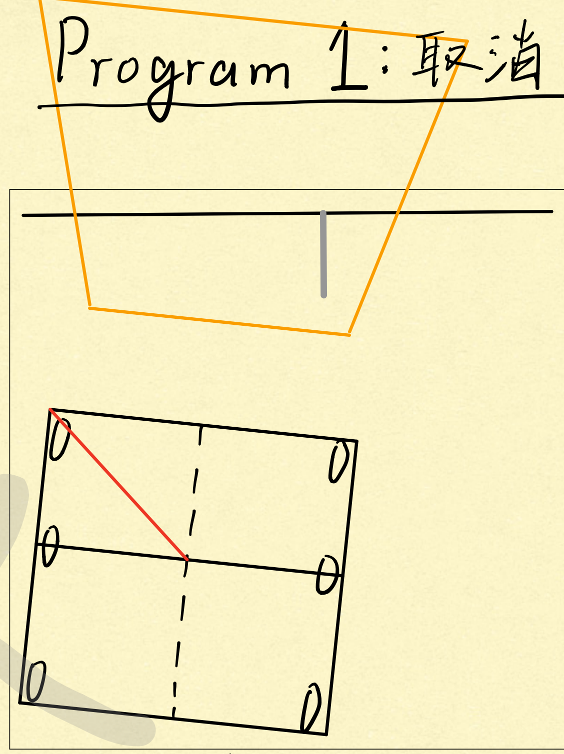
1. 摄像头竖直朝下拍摄，向前倾斜30度（右上角图片不代表拍摄到的图片，只用来辅助想象拟合后坐标系）
2. 树莓派以每秒约10帧的最大速度分析图像，拟合直线，将 $y = k \cdot x + b$ 中的 k、b 值以及摄像头拍到的3M反光条长度传给Java那边进行进一步处理
3. 直角坐标系的横轴是y轴，竖轴是x轴，坐标原点在图像的左上方。

（2019年03月24日拍摄，设备：Microsoft相机，于机器人顶部远离机械臂一侧）

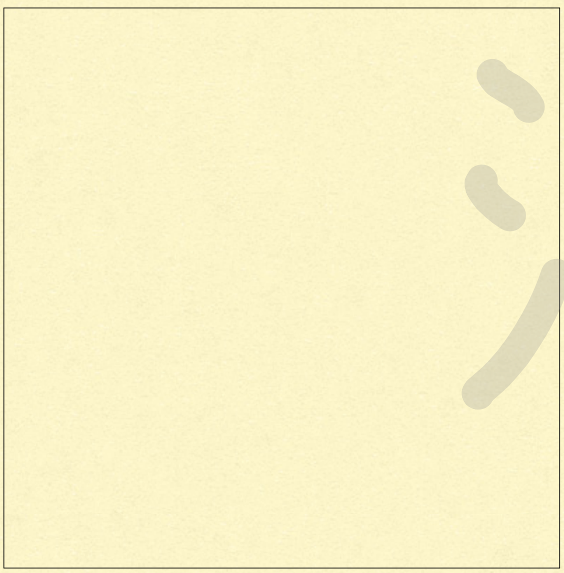
Program 1: 取消



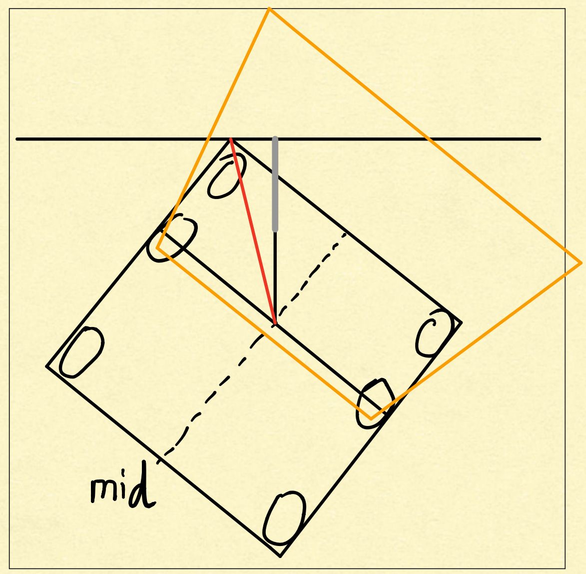
Step 1
(Might not be necessary ocassionally)



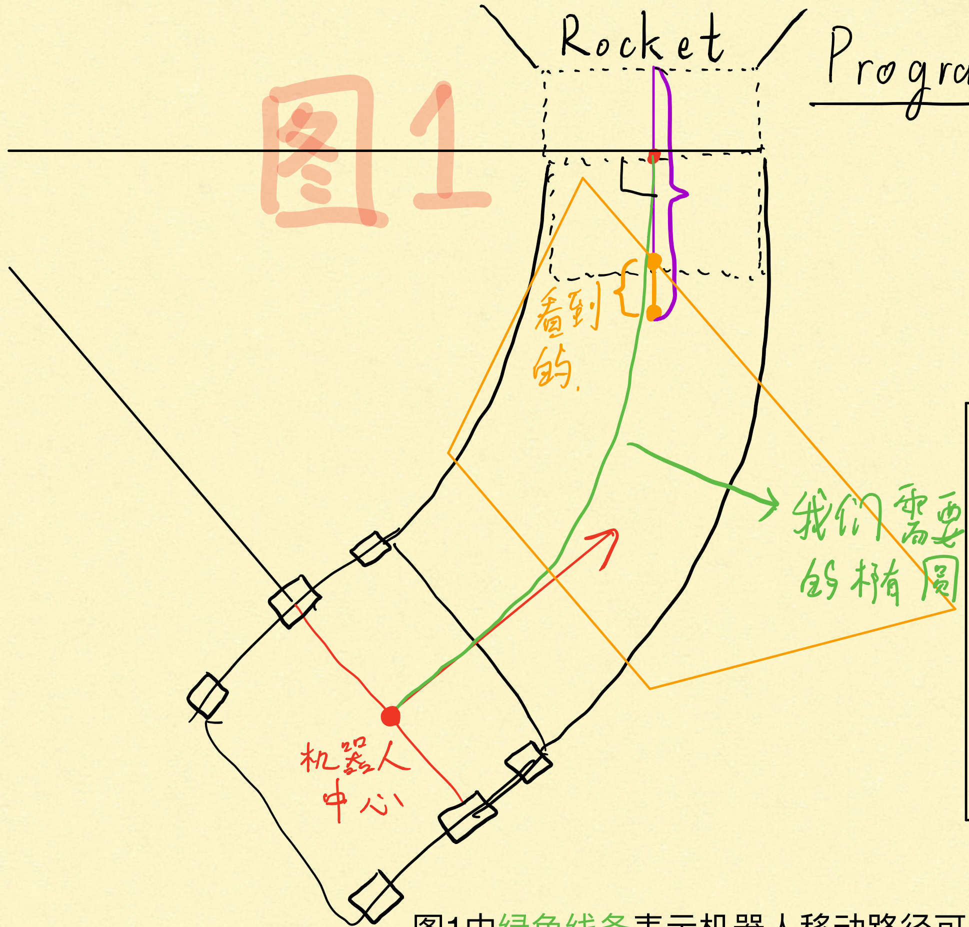
Step 2



Step 3



Step 4

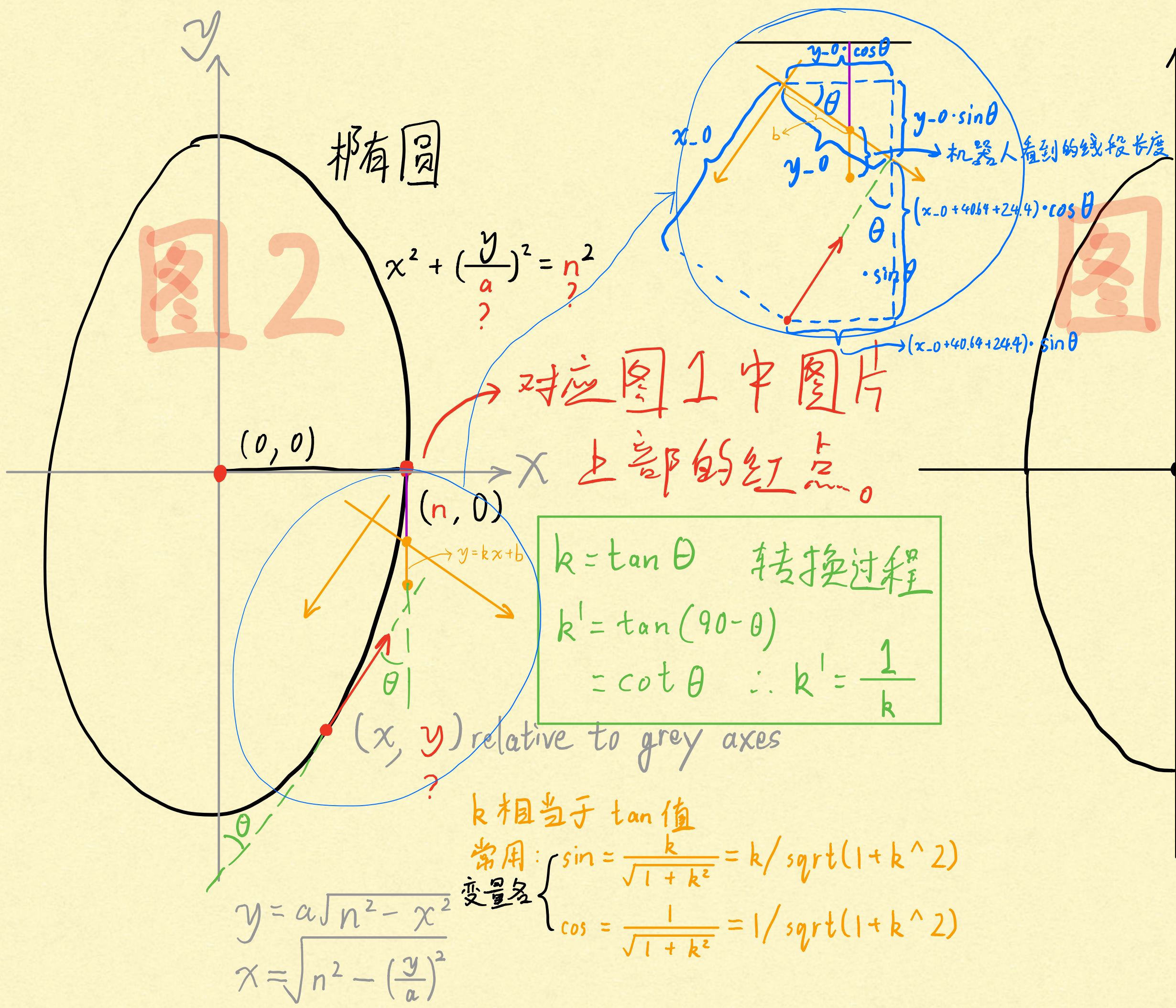


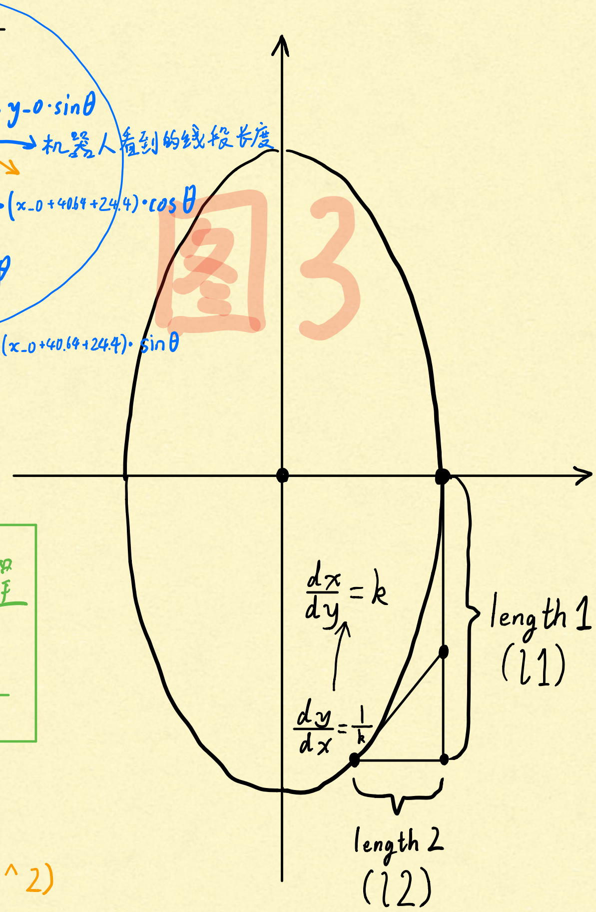
Program 2: (Complicated but Efficient)

NeedMeasure:
反光条总长度第一天上场后派人去量，把数据告诉Felix刘潇宇，让他把这个长写到Robot.java文件里面。

橙：看到的数据
↓
红：先要算的数据
↓
绿：最终要的数据

图1中绿色线条表示机器人移动路径可以用椭圆（$x^2+(ay)^2=n^2$）的第四象限来近似。用椭圆来近似是因为椭圆能够保证路径尽可能短，且最终机器人到Rocket放球的时候，传送带方向垂直于Rocket墙面，即便是中途受到干扰，也能够保证算法能够实时调整，提高了算法的Robustness。





图像分析与算法构建：（注：橙色坐标系中手写的x、y以下均用x_o、y_o表示，避免与x和y混淆）

第一步：图中从机器人中心红点位置（图2最靠下的一个红点）的椭圆切线，相对于灰色坐标系而言，椭圆切线斜率通过橙色坐标系转换过程，为1/k。

第二步：机器人相对于橙色坐标系的坐标是固定的，机器人中心相对于橙色坐标系的坐标为：

（x_o + 40.64 + 24.4，y_o）（不在代码中单独计算）（其中x_o、y_o）为下面量到的两个真实距离。

Calibration: 机器人前梁到机器人中轮轴心距离40.64cm; 机器人摄像头中心到机器人前梁距离24.4cm

（**代码开头要测量的部分**——将Microsoft摄像头连接到Windows电脑，打开“相机”，使模拟反光条的下方端点出现在画面正中央（如左图）。连接到树莓派，查看此时Felix写的算法返回的反光条长度值，在地上标记下模拟反光条下方端点的位置。然后调整反光条位置，使Felix写的算法返回的反光条长度值为0，再次在地上标记下模拟反光条下方端点的位置。测量地上两个标记点之间的实际距离(cm为单位)，除以第一次的Felix写的算法的返回值，即可得出Felix的算法测量的反光条长度值需要乘以的系数（来获得真实长度）。接着，用相似方法测量出图像中模拟反光条顶端到b=0位置时的真实距离，然后计算出系数。这两个系数的目的是为了校准本文档写作者所建立的数学模型与Felix建立的数学模型之间的长度统一性。（本文档模型使用与真实值相同的长度数值）

第三步： 计算图 3 中几个关键的数值。这些数值在之后的导数方程的求解中需要用到。

l1 = 反光条总长度 + y_o · sin + (x_o + 40.64 + 24.4) · cos - b_o · sin - 机器人看到的线段长度

l2 = b_o · cos - y_o · cos + (x_o + 40.64 + 24.4) · sin

（以上两个等式由图 2 、图 3 分析得出，且需要在代码部分进行计算以及变量赋值以节省性能）

第四步： 目的分析——目前需要的是椭圆的表达式，通过求解导数方程：x’ = dx/dy = k（该方程由图 3 分析得出），即可求出椭圆表达式中的两个关键参数—— a、n

第五步： 中间未知数分析。分析结果：共有三个值：y、a、n

第六步： 对椭圆表达式关于y微分。微分结果：$\boxed{x' = \frac{dx}{dy} = \frac{-2y}{a^2\sqrt{n^2 - \left(\frac{y}{a}\right)^2}}}$

第七步：解方程组：

$$\begin{cases} 2y + ak\sqrt{a^2n^2 - y^2} = 0 \\ n - length2 - \sqrt{n^2 - \left(\frac{y}{a}\right)^2} = 0 \\ y - length1 = 0 \end{cases}$$

中间过程：

y = length 1 (by using equation 3)

$$n - length2 - \sqrt{n^2 - \left(\frac{length1}{a}\right)^2} = 0$$

$$\cancel{n^2} - 2n\,length2 + (length2)^2 = \cancel{n^2} - \left(\frac{length1}{a}\right)^2$$

$$(length1)^2 - 2a^2n\,length2 + a^2(length2)^2 = 0$$

交换位置 ~~$a^2\cdot((length2)^2 - 2n\,length2$~~

$$a^2\cdot\left((length2)^2 - 2n\,length2\right) = (length1)^2$$

$$a^2 = \frac{(length1)^2}{2n\,length2 - (length2)^2}$$

$$a = \sqrt{}$$

(by using equation 2)

↓以下都是by using eq 1

$$2length1 + \sqrt{\frac{(length1)^2}{2n\cdot length2 - (length2)^2}}\cdot k\cdot\sqrt{\frac{n^2\cdot(length1)^2}{2n\,length2 - (length2)^2} - (length1)^2} = 0$$

$$2length1 + k\cdot\sqrt{\frac{(length1)^2}{2n\cdot length2 - (length2)^2}\cdot\left(\frac{n^2\cdot(length2)^2}{2n\,length2 - (length2)^2} - (length1)^2\right)} = 0$$

$$\frac{\cancel{l1^2}}{2n\cdot l2 - l2^2}\cdot\left(\frac{n^2\cdot l2^2}{2n\cdot l2 - l2^2} - l1^2\right) = \frac{4\cancel{l1^2}}{k^2}$$

$$\frac{n^2\cdot l2^2}{2n\cdot l2 - l2^2} - l1^2 = \frac{4(2n\cdot l2 - l2^2)}{k^2}$$

$$n^2\cdot l2^2 - (2n\cdot l2 - l2^2)\cdot l1^2 = \frac{4}{k^2}(2n\cdot l2 - l2^2)^2$$

$$l2^2\cdot n^2 - 2\cdot l1^2\cdot l2\cdot n + l1^2\cdot l2^2 = \frac{4}{k^2}(4\cdot l2^2\cdot n^2 - 4\cdot l2^3\cdot n + l2^4)$$

$$\frac{16}{k^2}\cdot l2^2\cdot n^2 - \frac{16}{k^2}\cdot l2^3\cdot n + \frac{4}{k^2}\cdot l2^4 - l2^2\cdot n^2 + 2\cdot l1^2\cdot l2\cdot n - l1^2\cdot l2^2 = 0$$

$$\left(\frac{16}{k^2} - 1\right)\cdot l2^2\cdot n^2 + \left(2\cdot l1^2\cdot l2 - \frac{16}{k^2}\cdot l2^3\right)\cdot n + \left(\frac{4}{k^2}\cdot l2^4 - l1^2\cdot l2^2\right) = 0$$

$$n = \frac{\frac{16}{k^2}\cdot l2^2 - 2\cdot l1^2\cdot l2^2 \pm\sqrt{\left(2\cdot l1^2 l2 - \frac{16}{k^2}\cdot l2^3\right)^2 - \left(\frac{64}{k^2} - 4\right)\cdot l2^2\cdot\left(\frac{4}{k^2}\cdot l2^4 - l1^2\cdot l2^2\right)}}{\left(\frac{32}{k^2} - 2\right)\cdot l2^2}$$

改动：代码中length1、length2全部改为l1、l2以统一。

结果：

y = length 1 (by using equation 3)

$$a^2 = \frac{(length1)^2}{2n\,length2 - (length2)^2}$$

$$a = \sqrt{}$$

$$n = \frac{\frac{16}{k^2}\cdot l2^2 - 2\cdot l1^2\cdot l2^2 \pm\sqrt{\left(2\cdot l1^2 l2 - \frac{16}{k^2}\cdot l2^3\right)^2 - \left(\frac{64}{k^2} - 4\right)\cdot l2^2\cdot\left(\frac{4}{k^2}\cdot l2^4 - l1^2\cdot l2^2\right)}}{\left(\frac{32}{k^2} - 2\right)\cdot l2^2}$$

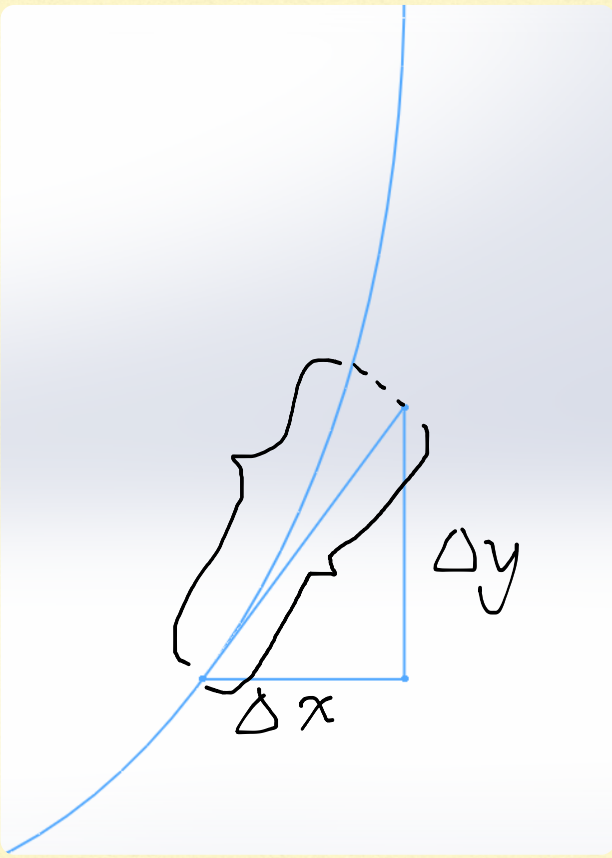
第八步：利用第七步中数学层面解出的 y、a、n 的表达式，在代码中创建 a、n 两个变量并赋值。

我们需要的是机器人每经过一小段距离时，它的机头朝向所需要变动的角度，因为有了这个角度才方便之后进行处理。所以第九步就是求单位经过路径机器人转动角度。

第九步：需要这个角度，就需要斜率的变化量；需要斜率的变化量，就需要代入机器人初始位置的 y 值，以及机器人最终位置的 y 值。初始位置的 y 值已有，最终位置的 y 值用初始位置的 y 值加0.001获得。

$$\Delta\theta = \arctan(x_2') - \arctan(x_1') \quad (\text{这个值应该是负的}) \quad rad$$

由于数学届没有发现椭圆弧长的计算公式，所以我们需要用方法来近似一段椭圆上的弧的长度，这里选用三角形来近似，理论上的误差值应该在 pi / 2 的倍率之内。



近似方法如图：
当三角形足够小的时候，图中标出的弧线段的长度会十分接近于三角形斜边的长度，则我们可以通过求出三角形斜边的长度，近似地对弧线段的长度进行估计。

为了使三角形足够小，代入极小的 Δy 值（$\Delta y = 0.01$）

$$\Delta l = \sqrt{\Delta x^2 + 0.01^2} = \sqrt{\left(\sqrt{n^2 - \left(\frac{y+0.01}{a}\right)^2} - \sqrt{n^2 - \left(\frac{y}{a}\right)^2}\right)^2 + 0.01^2}$$

这样需要的变量就都有了：
1. 角度的变化量 $\Delta\theta$
2. 斜率变化所对应的 Δy 所带来的椭圆弧上移动的长度近似值 Δl

$\frac{\Delta\theta}{\Delta l}$ 就是我们需要的单位经过路径机器人转动角度。

上面的 $\frac{\Delta\theta}{\Delta l}$ 还不是左右轮的转速之比，还需要最后一步处理，也就是下面的第十步。

第十步：
当角度的变化量足够小的时候，电机所导致的移动误差甚至会大于算法上近似所导致的移动误差。在此情况下，机器人的行进路径长度可以被近似为机器人左右两轮行进路径长度的平均值。由此前提假设，我们就可以将_为了使机器人转动 $\Delta\theta$ 而让左右轮产生的行进轨迹长度差_的其中一半在靠近圆心的轮子上减去，同时在远离圆心的轮子上加上，就能够实现让机器人在模型中的质心经过 Δl 的过程中逐渐同时形成 $\Delta\theta$ 的转向。

轮子到质心之间的距离，也是半径

让左右轮产生的行进轨迹长度差 $\Delta s = \Delta\theta \cdot 33.27785\,cm$

最终需要输出的左右轮转速比例 = $\frac{\omega_s}{\omega_L} = \frac{\frac{V_s}{r_{轮半径}}}{\frac{V_L}{r_{轮半径}}} = \frac{\frac{S_s}{t \cdot r_{轮半径}}}{\frac{S_L}{t \cdot r_{轮半径}}} = \frac{S_s}{S_L} = \frac{\Delta l - \Delta s}{\Delta l + \Delta s}$

$\Delta l - \Delta s$ → 靠近圆心的轮子

$\Delta l + \Delta s$ → 远离圆心的轮子