

第四课 机器人运动学模型的搭建

1. 两轮差分模型的机器人

随着高技术的发展,机器人技术的应用领域不断扩大,工业机器人、特种作业机器人、服务机器人、微小型机器人等已经在各个方面得到广泛的应用和发展。两轮差速机器人涉及到多个应用领域,对两轮差速机器人的运动控制研究成为主要的课题。对轮式移动机器人,做路径和轨迹时,一般是以独立驱动轮的轮基中点为基点来进行的。在实时控制中,轨迹产生器把规划好的路径转变成随时间变化的两个独立驱动轮的角速度,通过控制器,分别去控制两个驱动轮。

机器人模糊控制有两方面的独特优势:一方面它简化了控制算法;另一方面可以用于探索模糊逻辑和分析方法学在改善控制系统性能方面的能力。机器人控制的主要任务是保证机器人实现有目标的运动。由于模糊控制作用具有模拟人脑逻辑推理和思维控制的特点,而智能机器人正是具有模拟或者代替人的部分功能,所以近年来模糊控制已经被广泛应用于智能机器人的控制中。

机器人的运动一定是在一定的坐标系中进行的,而 ROS 使用的是右手坐标系,所以我们需要首先了解一下右手坐标系。右手坐标系如下图 1-1-0 所示,右手握拳,大拇指指向了 Z 轴,食指指向了 X 轴,中指指向了 Y 轴,对应到我们在平面运动的机器人,运动控制就是控制其在世界坐标系的 XY 轴平面的运动,Z 轴方向可以看做是静止的,也就是在运动过程中 Z 轴方向的线速度永远为 0,从机器人坐标系来看 Y 轴方向的线速度也为 0,但角速度则是反过来的,Z 轴方向的角速度不为 0,而 XY 轴方向的角速度都为 0。

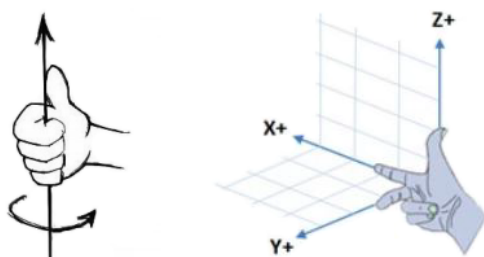


图 1-1-0 右手笛卡尔坐标系

2. HiBot 运动学正解

在机器人底层驱动程序的开发中,经常用到航迹推演 (Odometry),无论是定位导航还是普通的方向控制。航迹推演中除了对机器人位姿进行估计,另一个很重要的关系是移动机器人前进速度 v 、转向角速度 ω 与左轮速度 v_l 、右轮速度 v_r 之间的转换。

在底层控制代码中,我们已经通过编码器采样计算出了两个轮子在某一周期内各自的速度,那么该如何将机器人轮子运动的速度转化为右手笛卡尔坐标系中的 X Y 的线速度,以及 Z 方向的角速度呢? 这个就是运动学模型的正解问题。

图 2-1-0 是移动机器人在两个相邻时刻的位姿,其中 θ_1 是两相邻时刻移动机器人绕圆弧运动的角度, θ_3 是两相邻时刻移动机器航向角 (朝向角 head) 的变化量。 l 是左右轮之间的间距, d 是右轮比左轮多走的距离。 r 是移动机器人圆弧运动的半径。

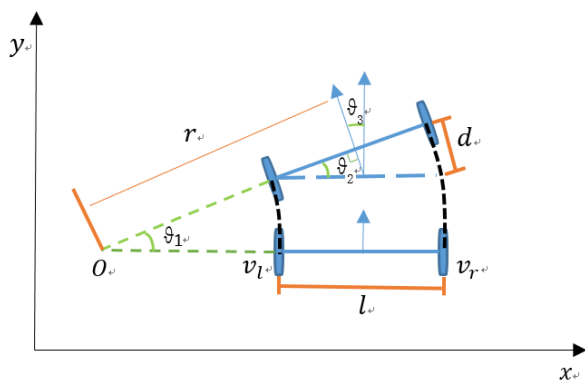


图 2-1-0 相邻时刻的位姿态

- (1) 移动机器人前进速度 v 等于左右轮速度的平均

$$v = \frac{v_r + v_l}{2}$$

- (2) 现在来推导机器人航向角如何计算，以及如何计算角速度 w 。如图 2-1-0 所示，把两个时刻的机器人位置叠加在一起，可以清楚的看到移动机器人航向角变化量是 θ_3 。从图中的几何关系可以得到：

$$\theta_3 = \theta_2 = \theta_1$$

- (3) 也就是说移动机器人航向角变化了多少角度，它就绕其运动轨迹的圆心旋转了多少角度。这句话很好验证，我们让机器人做圆周运动，从起点出发绕圆心一圈回到起点处，在这过程中机器人累计的航向角为 360 度，同时它也确实绕轨迹圆心运动了 360 度，说明机器人航向角变化多少度，就绕圆心旋转了多少度。而这三个角度中，很容易计算出来，由于相邻时刻时间很短，角度变化量很小，有下面的近似公式：

$$\theta_2 \approx \sin(\theta) = \frac{d}{l} = \frac{(v_r - v_l) \cdot \Delta t}{l}$$

所以可以得到机器人绕圆心运动的角速度 w ，它也是机器人航向角变化的速度：

$$w = \frac{\theta_1}{\Delta t} = \frac{v_r - v_l}{l}$$

2.1 用代码实现运动学正解的过程

```
Send_Data.Sensor_Str.X_speed = (Left_moto.Current_Speed + Right_moto.Current_Speed)/2.0f;
Send_Data.Sensor_Str.Y_speed = 0.0;
Send_Data.Sensor_Str.Z_speed = (Right_moto.Current_Speed - Left_moto.Current_Speed)/Base_Width;
```

3. HiBot 运动学逆解

当知道了正解的过程，那么也就大概知道什么是逆解了，运动逆解过程就是给定基于右手坐标系的 X 方向的线速度和 Z 方向的角速度，该如何求解两个轮子的转速。运动学逆解的过程我们不再一一推导，直接给出最终代码：

```
/*
@ description: Chassis kinematics Positive solution function
@ param: float vx,float vz
@ return: none
@ author: Xuwei Zhou
@ date : 2019-4-17
@ note:
@ function: void Kinematics_Positive(float vx,float vz)
*/
```

```
void Kinematics_Positive(float vx, float vz)
{
    if(vx == 0.0f){           //原地旋转或静止
        Right_moto.Target_Speed = vz * Base_Width / 2.0f;
        Left_moto.Target_Speed  = (-1) * Right_moto.Target_Speed;
    }
    else if(vz == 0.0f){      //静止或者前后运动
        Right_moto.Target_Speed = Left_moto.Target_Speed = vx;
    }
    else{                     //在前进或者后退过程中转弯
        Left_moto.Target_Speed  = vx - vz * Base_Width / 2.0f;
        Right_moto.Target_Speed = vx + vz * Base_Width / 2.0f;
    }
}
```

4. 总结

本节课程我们学习了什么是运动学模型,然后选定最经典的两轮差分机器人的运动学模型进行推导和分析。同时理解了什么是运动学正解,什么是运动学逆解,最终转化为我们可以使用的代码。