

# 智能无人系统综合设计

## 前言

通过对智能无人系统综合设计（II）课程的学习，我收获了很多知识干货，本篇文章主要是对在课程的理论讲解和实践过程中获得的一些知识的总结。在本次课程中我了解到了四旋翼无人机（以大疆RoboMaster TT为例）和两轮差速机器人（以天之博特TIANBOT MINI机器人为例）的工作原理，掌握了如何使用ROS机器人操作系统对他们进行分别控制和协同控制。

## 一、硬件简介

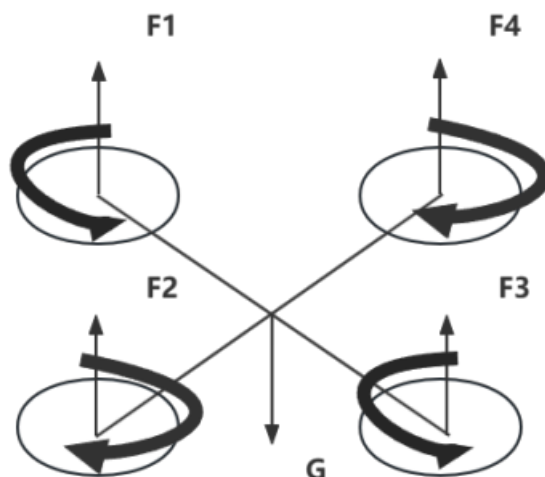
### 1. 无人机

RoboMaster TT大疆无人机



#### 1.1 机翼

四旋翼无人机，3英寸桨叶。四旋翼机身同一对角线上的两个旋翼归为一组，通过控制对称的四个旋翼的转速，形成相应不同的运动组合实现各个飞行状态的控制，包括悬停、垂直运动、翻滚运动、俯仰运动以及偏航运动五种状态。



在悬停状态下，由于无人机的四个旋翼具有相同的转速，产生的上升合力正好与自身重力相等；并且因为旋翼转速大小相同、前后端和左右端转速方向相反，从而使得无人机总扭矩为零，得以静止在空中，实现悬停状态。

在保证四旋翼无人机每个旋翼转速大小相同、前后端和左右端转速方向相反的情况下，同时对每个旋翼增加/减小大小相同的转速，便可实现无人机的垂直运动。

翻滚运动是在保持四旋翼无人机前后端旋翼转速不变的情况下，通过改变左右端的旋翼转速，使得左右旋翼之间形成一定的升力差，从而使得沿无人机机体左右对称轴上产生一定力矩，导致在方向上产生角加速度实现控制。

与翻滚运动相似，在保持四旋翼无人机左右端旋翼转速不变的情况下，通过改变前后端的旋翼转速，形成前后旋翼升力差，从而在机体前后对称轴上形成一定力矩，引起角方向上的角加速度实现控制。

保持前后端或左右端旋翼转速相同时，其便不会发生俯仰或翻滚运动；而当每组内的两个旋翼与另一组旋翼转速不同时，由于两组旋翼旋转方向不同，就会导致反扭矩力的不平衡，此时便会产生绕机身中心轴的反作用力，引起沿角角加速度。当前后端旋翼的转速相同并大于左右端旋翼转速时，因为前者沿顺时针方向旋转，后者相反，总的反扭矩沿逆时针方向，反作用力作用在机身中心轴上沿逆时针方向，引起逆时针偏航运动；反之，则会引起顺时针偏航运动。

## 1.2 相机

500万像素相机，支持JPG（照片）和MP4（视频），支持电子防抖。无人机通过软件系统对于图像的晃动进行自动微调，同时也可以通过对比相邻帧之间的差异来去除震动。

## 1.3 拓展模块



开源控制器，8\*8红蓝双色可编程点阵LED。开源控制器由程序计数器、指令寄存器、指令译码器、时序产生器和操作控制器组成，按照指令控制无人机的运动。点阵使用同一时刻只有一列或一行有LED显示的扫描显示方式，利用了人的视觉暂留效果，当扫描足够快的时候，人眼是看不到闪烁从而看到完整的显示信息。

## 2. 两轮差速无人车

天之博特TIANBOT\_MINI两轮差速无人车



### 2.1 两轮差速控制

两轮差速无人车底盘由两个动力轮位于底盘左右两侧，两轮独立控制速度，通过给定不同速度实现底盘转向控制。

运动轨迹推算公式：

$$\begin{aligned}x_n &= x_{n-1} + \Delta x = x_{n-1} + \Delta d * \cos\theta \\y_n &= y_{n-1} + \Delta y = y_{n-1} + \Delta d * \sin\theta\end{aligned}$$

### 2.2 二维激光雷达

该无人车搭载360°2D激光雷达，由激光发射器、接收器、信号处理单元和电机组成，激光器发射激光信号到被测物体，经反射后到达光电器件，进行信号转换给到回波信号处理电路，最终获得电脉冲信号传给计时器，计算出光飞行的时间，从而计算出雷达到被测物体的距离。

## 二、ROS机器人操作系统基础使用

### 1. ROS节点和launch文件

#### 1.1 ros工作空间与功能包

ros工作空间简单来说相当于工程或者项目，编写ros程序之前，首先需要创建ros工作空间。创建工作空间的步骤：

```
1 | mkdir -p workspace/src
2 | cd workspace
3 | catkin_make
```

ros功能包用于区分某个文件夹下代码的功能，如机器人有移动，导航，视觉等功能，我们为整个机器人创建工作空间，也可称为一个项目，接下来要开发它的各个功能，于是我们使用功能包来分隔一类代码。创建功能包的步骤：

```
1 | cd workspace/src
2 | catkin_create_pkg package rospy
```

#### 1.2 rospy创建ROS节点

ros最小的进程单元就是节点，一个ros功能包里可以有多个可执行文件，可执行文件在运行之后就成了一个进程，这个进程就叫做节点。使用rospy创建ROS节点的步骤：

```
1 cd package
2 mkdir scripts
3 cd scripts
4 vm rosnode.py
5 chmod +x rosnode.py
```

`rospy.init_node` 函数用于创建该节点的名字。

### 1.3 集成ROS节点的launch文件

许多ROS软件包都带有启动文件，即launch文件，这些启动文件通常会为软件包提供一组节点，这些节点聚合了一些功能，通过roslaunch命令同时启动这些节点。

launch文件采用XML格式书写，文件标签共11种，标签按顺序进行浏览，标签值取最后的设定。

## 2. ROS通信机制

ROS节点之间主要有话题通信和服务通信两种方式，本次课程主要以话题通信为主。

### 2.1 发布话题

发布者（Talker）启动，通过RPC向 ROS Master 注册发布者的信息，包括：发布者节点信息，话题名，话题缓存大小等；Master 会将这些信息加入注册列表中。

### 2.2 接收话题

订阅者（Listener）启动，通过 RPC 向 ROS Master 注册订阅者信息，包括：订阅者节点信息，话题名等；Master 会将这些信息加入注册列表。

### 2.3 回调函数

ROS中回调函数的主要作用是用来处理被调用的数据，其原理是除了用户的主程序以外，ROS的socket连接控制进程会在后台接收订阅的消息，所有接收到的消息并不是立即处理，而是等到spin()或者spinOnce()执行时才集中处理。所以为了保证消息可以正常接收，需要尤其注意spinOnce()函数的使用。

## 3. TF坐标变换

一个机器人中通常有多个坐标系，如世界坐标系、基座坐标系、夹具坐标系等等，这些坐标系在tf中构成坐标树，tf通过坐标树维护多个坐标系之间的坐标变换，ros坐标系可以在分布式系统中进行变换，这意味着坐标对所有的节点都是可用的，同时所有订阅tf消息的节点都会缓存一份所有坐标系的变换关系数据，而不需要中心节点来存储。

### 3.1 数据类型

数据类型定义模块：`import tf` 或 `import tf2`

该头文件中含有 `Quaternion`，`Vector`，`Point`，`Pose`，`Transform` 等类型，这些数据类型中有些是结构体类型，有些是模板类。

### 3.2 广播与监听

广播：向系统中广播坐标系之间的坐标变换关系。系统中可能存在多个不同部分的TF变换广播，每个广播都可以直接将坐标变换关系插入TF树中，不需要再进行同步

监听：接收并缓存系统中发布的所有坐标变换数据，并从中查询所需要的坐标变换关系。

## 4. 机器人模型描述语言

### 4.1 urdf

urdf是ROS用于描述机器人所有元素的模型文件，采用XML格式书写。

## 4.2 xacro

xacro是升级版的urdf，简化了urdf模型，用于复杂机器人的建模。对比urdf，xacro精简模型代码，能够创建宏定义和文件包含关系，提供可编程接口，具有常量、变量、数学计算、条件语句等语法。

## 三、任务规划

### 1. 有限状态机

smach是状态机的意思，是基于Python实现的一个功能强大且易于扩展的库。smach本质上不依赖于ROS，可以用于任意python项目，不过在ROS中元功能包executive\_smach将smach和ROS很好的集成在了一起，可以为机器人复杂应用开发提供任务级的状态机框架，此外元功能包还集成了actionlib和smach\_viewer。

#### 1.1 功能

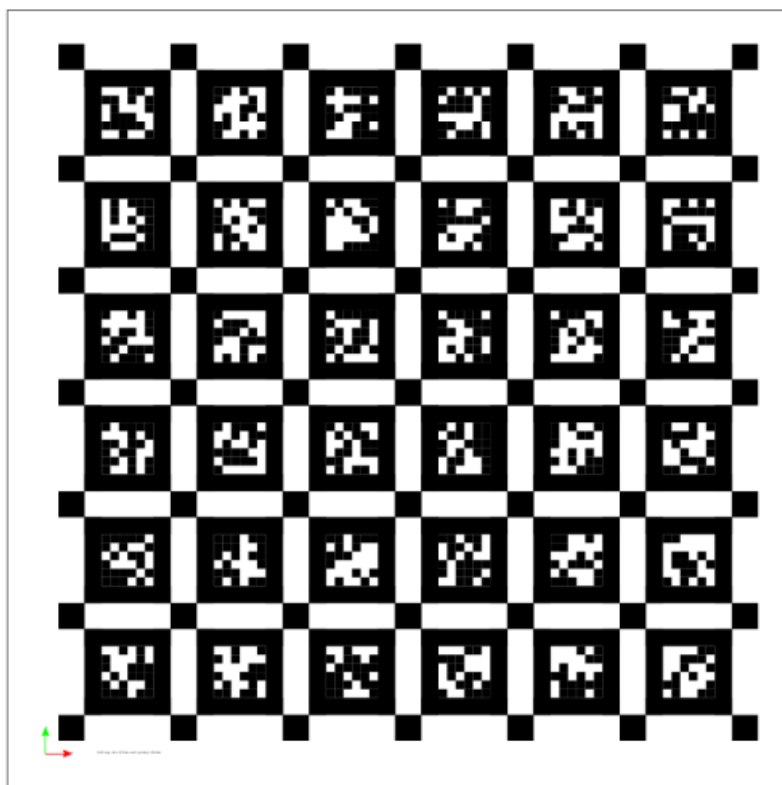
在很多应用场景中，我们需要设计一些复杂的机器人任务，任务中包含多个状态模块，而这些状态模块（State）之间在某些情况下会发生跳转，这就是smach可以发挥作用的地方。

#### 1.2 优点

smach适合快速原型设计，支持设计、维护、调试大型复杂的状态机，同时提供可视化观测工具smach\_viewer，可以看到完整状态机的状态跳转、数据流等信息。

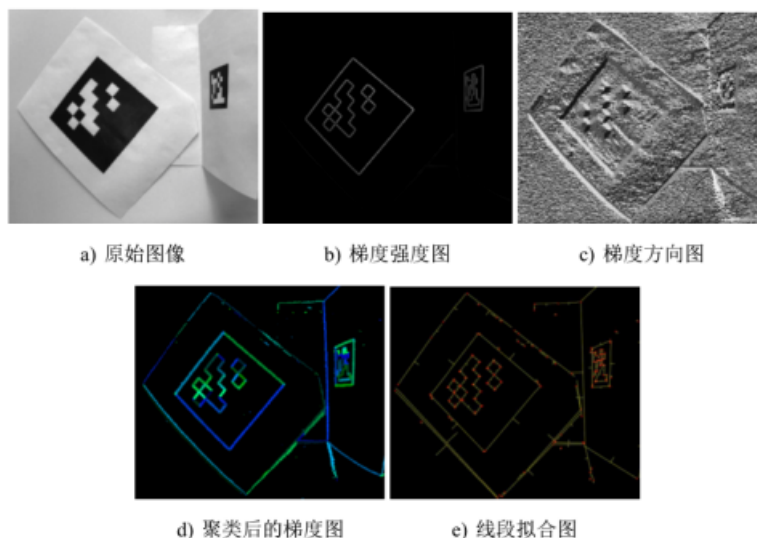
### 2. apriltag二维码检测

apriltag二维码用于机器视觉中，可以简单理解为是简化版的二维码，具有识别距离远、识别速度快、识别准确的特点。其主要用途包括相机标定、目标大小估计、单目距离测量等。



#### 2.1 原理

二维码的检测工作的思路是提取一块四边形的区域，这个区域具有比周围亮度更低的特点。这个思路的主要优点是尽可能多的检测出所有可能的二维码，但是极有可能出现 false positive，后续的编码可以对野值进行剔除。二维码检测步骤如下图：



与其他的二维码检测程序相似，首先会计算整个图像上每个像素的梯度的强度和方向，然后对梯度的强度和方向进行聚类。聚类算法是基于图的算法，具体计算公式如下：

$$D(n \cup m) = \min(D(n), D(m)) + K_D / |n \cup m|$$

$$D(n \cup m) = \min(M(n), M(m)) + K_M / |n \cup m|$$

检测得到所有的线段后，下一步需要挑选线段，对线段分组，每一组线段都是能够构成四边形的候选者。分组遵循以下规则：前一条边的末端点应该和下一条边的始端点之间的距离小于一个阈值，并且相接的线段要构成逆时针的方向。

二维码的位置和姿态的计算需要额外的信息：相机的焦距和二维码的物理距离。3×3 的单应矩阵可以写成3×4的相机内参矩阵P和4×3的外参矩阵E的积。外参矩阵通常是4×4的，但是在二维码坐标系中的四个角点的坐标在z方向为0，所以就可以移除外参矩阵的第三列元素，使用旋转参数和平移参数来表示截取后的外参矩阵，那么单应矩阵就可以表示为：

$$\begin{bmatrix} h_{00} & h_{01} & h_{02} \\ h_{10} & h_{11} & h_{12} \\ h_{20} & h_{21} & h_{22} \end{bmatrix} = sPE = \begin{bmatrix} f_x & 0 & 0 & 0 \\ 0 & f_y & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} R_{00} & R_{01} & T_x \\ R_{10} & R_{11} & T_y \\ R_{20} & R_{21} & T_z \end{bmatrix}$$

由于单应矩阵是两组齐次坐标的映射关系，所以存在一个尺度参数s，将上式展开，得到：

$$h_{00} = sR_{00}f_x$$

$$h_{01} = sR_{01}f_x$$

$$h_{02} = sT_xf_x$$

在二维码的解码过程中，首先将二维码中每一块的坐标通过单应性矩阵映射到图像平面，判断图像平面上映射后的点的像素值是否大于某一个阈值，若大于该阈值，二维码这个坐标上为1，相反，小于该阈值则判定为0。阈值的选取为了满足光照不变性，定义了以下的阈值模型：

$$I(x, y) = Ax + Bxy + Cy + D$$

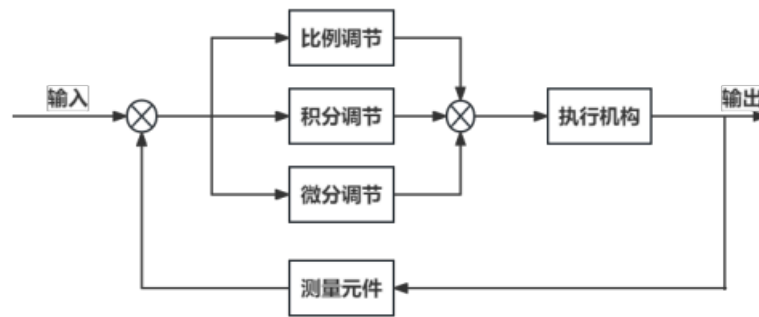
## 2.2 应用

apriltag二维码主要用于相机标定、目标大小估计、测量相机到目标的距离、3D定位、机器人SLAM、自主导航等领域。

## 3. PID控制



PID控制，即Proportional (P)- Integral(I) - Derivative(D) Control, 实际上是三种反馈控制：比例控制，积分控制与微分控制的统称。根据控制对象和应用条件，可以采用这三种控制的部分组合，即P控制，PI控制，PD控制或者是三者的组合，即真正意义上的PID控制。我们可以笼统地去称呼他们为PID控制律。采取这种控制规律的控制器，我们称为PID控制器。



### 3.1 原理

比例（P）控制：比例控制是一种最简单的控制方式。其控制器的输出与输入误差信号成比例关系。当仅有比例控制时系统输出存在稳态误差（Steady-state error）。

积分（I）控制：在积分控制中，控制器的输出与输入误差信号的积分成正比关系。对一个自动控制系统，如果在进入稳态后存在稳态误差，则称这个控制系统是有稳态误差的或简称有差系统（System with Steady-state Error）。

微分（D）控制：在微分控制中，控制器的输出与输入误差信号的微分（即误差的变化率）成正比关系。自动控制系统在克服误差的调节过程中可能会出现振荡甚至失稳，解决的办法是使抑制误差的作用的变化“超前”。

### 3.2 应用

PID控制适用于需要精确控制的各种自动控制系统，包括温度控制、机械控制、液位控制、流量控制等领域，其简单易用、性能稳定，因此在实际生产生活中得到了广泛的应用。

## 4. 霍夫圆变换

霍夫圆变换是将二维图像空间中一个圆转换为该圆半径、圆心横纵坐标所确定的三维参数空间中一个点的过程，因此，圆周上任意三点所确定的圆，经霍夫变换后在三维参数空间应对应一点。

### 4.1 原理

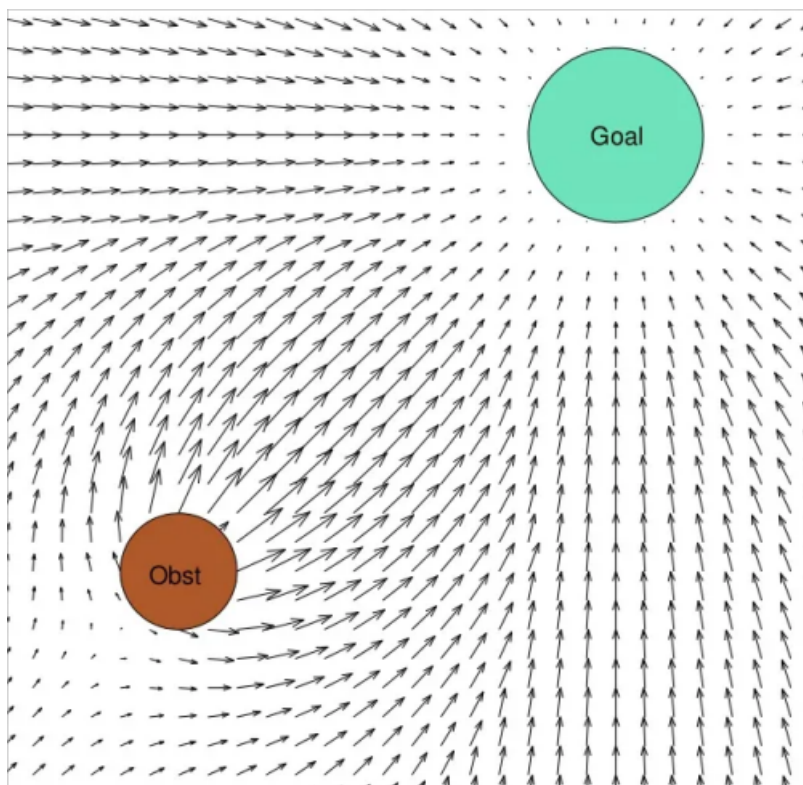
- (1) 首先对图像应用边缘检测，比如用canny边缘检测。
- (2) 使用sobel算子计算所有像素的梯度。
- (3) 遍历canny之后的所有非0的像素点，沿着梯度方向画线，每个点有是一个累加器，有一个线经过该点，累加器加1，对所有累加器进行排序，根据阈值找到所有可能的圆心。
- (4) 计算canny图像中所有的非0像素点距离圆心的距离，距离从小到大排序，选取合适的半径。
- (5) 对选取的半径设置累加器，对于满足半径r的累加器+1。

### 4.2 应用

霍夫圆变换是一种非常实用的图像处理技术，可以在许多领域中应用，如计算机视觉、医学影像、工业检测、计算机辅助设计和生物医学等。

## 5. 势场法路径规划

人工势场法是一种经典的机器人路径规划算法。该算法将目标和障碍物分别看做对机器人有引力和斥力的物体，机器人沿引力与斥力的合力来进行运动。



### 5.1 原理

人工势场方法背后的基本思想是，把机器人当作位形空间中处于人工势场影响下的一个质点。机器人能够被最终位形吸引到；同时会受到边界的排斥而远离障碍区域。

某点处的势函数表达为引力势和斥力势之和，势场构造如下：

$$U(q) = U_{att}(q) + U_{rep}(q)$$

路径规划问题可以被转化为一个数值优化问题，即以初始位形作为起始，寻找势场的全局最小值，解决该问题的最简单算法之一是梯度下降法，在这种情况下，势场的负梯度可被认为是位形空间中作用在机器人上的一个广义力：

$$F = -\nabla U$$

作为质点的机器人受到引力与斥力的作用。

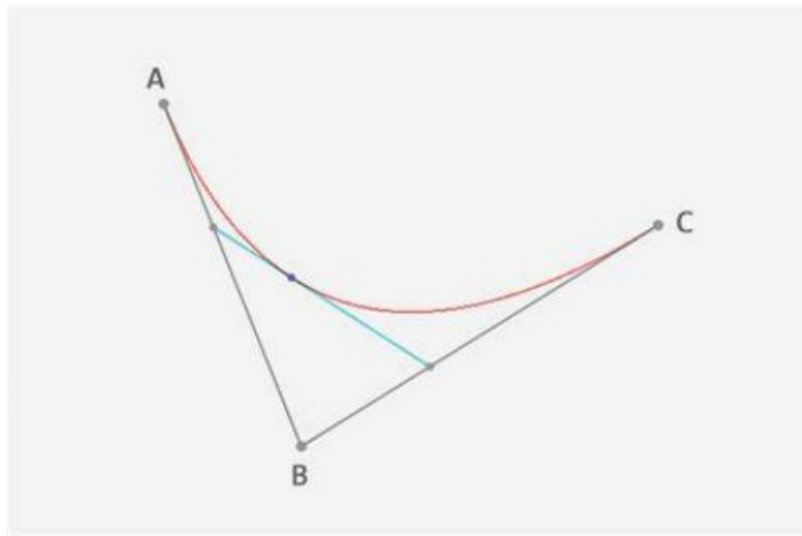
### 5.2 应用

人工势场法因模型设计简单、规划效率高，被广泛应用于不同场景下的地面机器人、机器船、无人飞行器等的路径规划。

## 6. Bezier曲线路径平滑

Bezier曲线是应用于二维图形应用程序的数学曲线，由线段与节点组成，节点是可拖动的支点，线段像可伸缩的皮筋。





## 6.1 原理

Bezier曲线公式：

$$p(t) = \sum_{i=1}^n P_i B_{i,n}(t) \quad 0 \leq t \leq 1$$

$$B_{i,n}(t) = C_n^i t^i (1-t)^{n-i}$$

$$C_n^i = \frac{n!}{i!(n-i)!} \quad (i = 0, 1, \dots, n)$$

Bernstein多项式递推性：

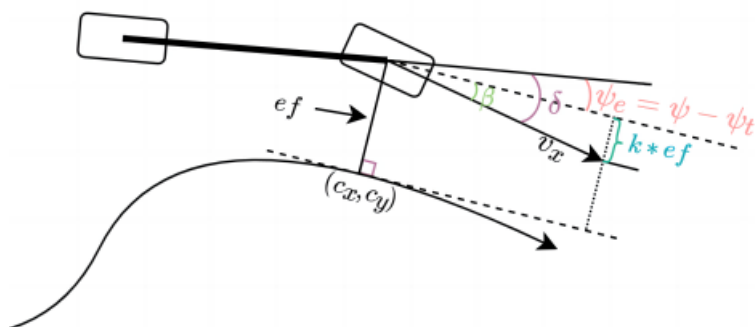
$$B_{i,n}(t) = (1-t)B_{i,n-1}(t) + tB_{i-1,n-1}(t) \quad (i = 0, 1, \dots, n)$$

## 6.2 应用

Bezier曲线广泛地应用于很多图形图像软件中，同时利用Bezier曲线进行路径规划中的路径平滑，能够使控制平台更好地进行位置转换。

## 7. Stanley路径跟踪

Stanley方法是一种基于横向跟踪误差的非线性反馈函数，并且能实现横向跟踪误差指数收敛于0。



## 7.1 原理

前轮转角控制变量由两部分组成，一部分是航向误差引起的转角，另一部分是横向误差引起的转角。

$$\delta = \psi_t - \psi + \arctan \frac{k e_y}{v}$$

横向误差的变化率为：

$$\dot{e}_y = -v \sin \delta_e$$

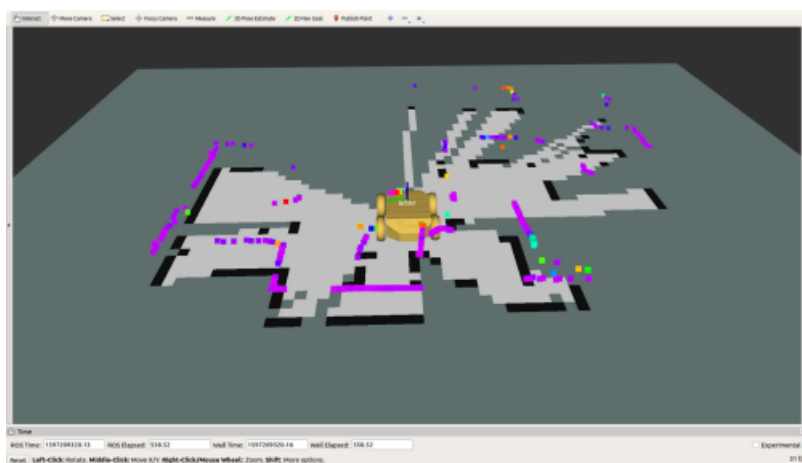
## 7.2 应用

stanley算法应用在控制平台的路径规划后的路径跟踪中，考虑到了预瞄点和真实车辆的转角偏差，使控制平台可以根据需求动态调整前馈控制增益系数，让实际运动更加和谐。

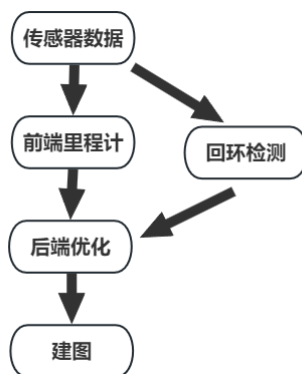
## 8. Slam二维激光雷达建图

SLAM是指运动物体根据传感器的信息，一边计算自身位置，一边构建环境地图的过程，解决机器人等在未知环境下运动时的定位与地图构建问题。激光SLAM是目前最稳定、最主流的定位导航方法。

激光SLAM所需要的传感器一般有激光雷达（Lidar）、惯性测量单元（IMU）、里程计（Odometry）。由于IMU具有较高的角速度测量精度，相比IMU里程计具有较高的局部位置测量精度，一般用IMU计算角度信息，里程计计算位置信息，配合激光雷达进行SLAM过程。



激光SLAM框架一般分为前端扫描匹配、后端优化、闭环检测、地图构建四个关键模块。



### 8.1 原理

目前SLAM算法大多采用标准的两步递归预测（时间更新）校正（测量更新）的形式：

$$P(X_k, m | Z_{0:k-1}, U_{0:k}, X_0) = \int P(X_k | X_{k-1}, u_k) * P(X_{k-1}, m | Z_{0:k-1}, U_{0:k-1}, X_0) dX_{k-1}$$

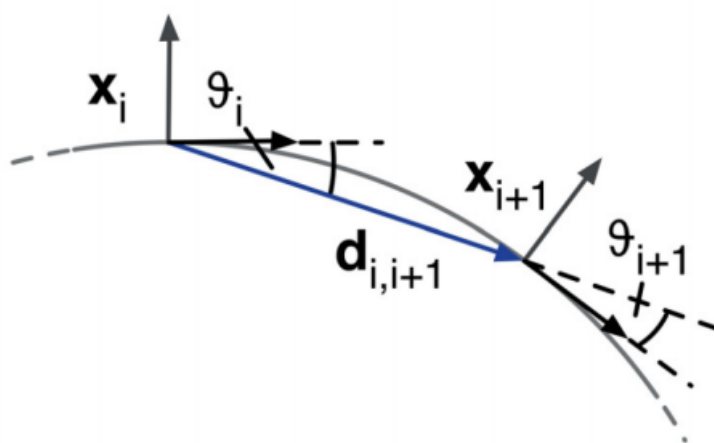
$$P(X_k, m | Z_{0:k}, U_{0:k}, X_0) = \frac{P(Z_k | X_k, m) P(X_k, m | Z_{0:k-1}, U_{0:k}, X_0)}{P(Z_k | Z_{0:k-1}, U_{0:k})}$$

### 8.2 应用

目前以SLAM导航技术为支撑的自主移动应用领域已经十分广泛了，涵盖航天、军事、特种作业、工业生产、智慧交通、消费娱乐等众多领域。航天领域的典型应用要属火星探测车，在遥远的星球上自主移动无疑是一项必备的技能。军事上借助自主移动的坦克、机器人士兵、飞机等，可以打一场无人化战争。在特种作业场合的自主移动机器人将发挥无可替代的作用，比如管道清洗、矿井作业、抢险救援、排爆、安防巡检、深海勘探等。农业上的应用，比如自主栽培、自主除草、自主施肥、自主采摘等。还有自动驾驶汽车、机器人终端物流配送、全自动化工厂、机器人智慧养老、机器人餐厅、家庭服务机器人等。

## 9. Teb路径规划

将给定的路径视为受内外力影响的弹性橡皮筋，使其变形，而内外力相互平衡，使路径收缩，同时与障碍物保持一定的距离，其中内外力就是对机器人运动的所有约束，而对于Teb算法，则在给定路径中间插入N个控制橡皮筋形状的控制点（机器人姿态），在点与点之间定义运动时间。



### 9.1 原理

惩罚函数：

$$f_{path} = e_{\tau}(d_{min,j}, r_{p_{max}}, \in, S, n)$$

$$f_{ab} = e_{\tau}(-d_{min,j}, -r_{O_{min}}, \in, S, n)$$

线速度约束：

$$f_{v_i} = e_{\tau}(v_i, v_{max}, \in, S, n)$$

运动学约束：

$$f_k(x_i, x_{i+1}) = \left\| \left[ \begin{pmatrix} \cos\beta_i \\ \sin\beta_i \\ 0 \end{pmatrix} + \begin{pmatrix} \cos\beta_{i+1} \\ \sin\beta_{i+1} \\ 0 \end{pmatrix} \right] \times d_{i,i+1} \right\|^2$$

最快路径约束：

$$f_k = (\sum \Delta T_i)^2$$

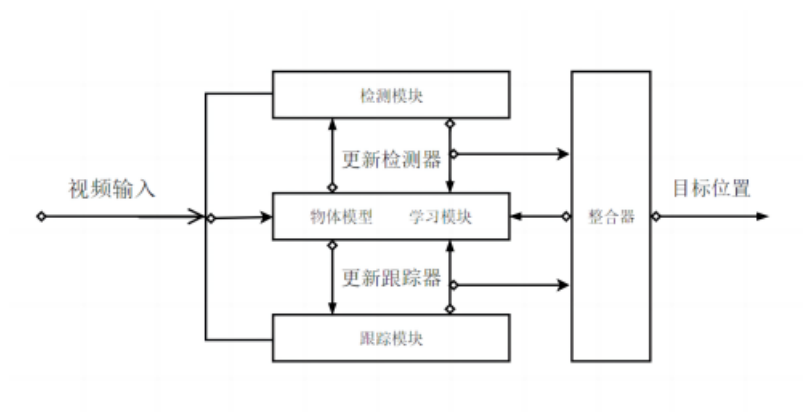
### 9.2 优点

Teb算法适合各种常见无人车模型，有很强的前瞻性，对前方一段轨迹进行优化，对动态障碍有较好的避障效果。

## 10. 其他

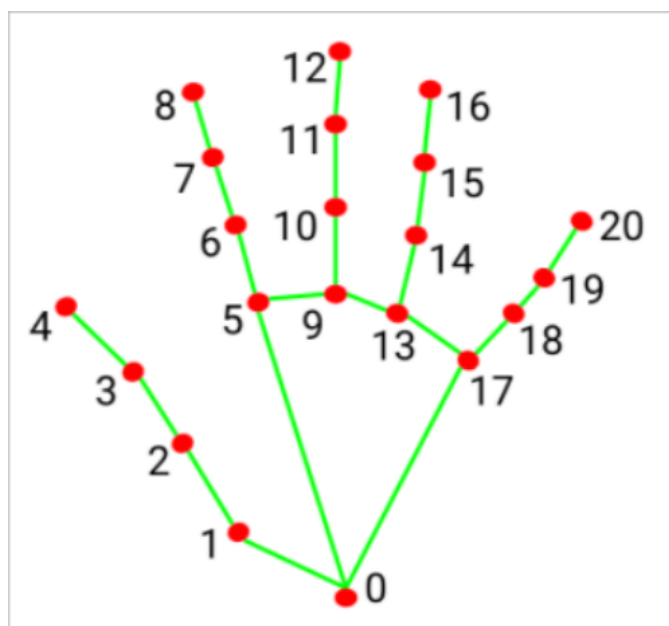
### 10.1 无人机通用追踪

TLD算法由三个模块组成:跟踪模块、学习模块和检测模块。检测模块和跟踪模块在学习模块的监督下并行工作。TLD的结构如下图所示。与传统的单目标跟踪算法相比,TLD跟踪算法融合了这三个模块,既能够对连续运动目标进行跟踪,又能够对遮挡再现目标重新检测跟踪,抗遮挡性能优越。在整个跟踪过程中,学习模块对检测器进行实时更新,能保持检测器的对目标形态变化的适应力。



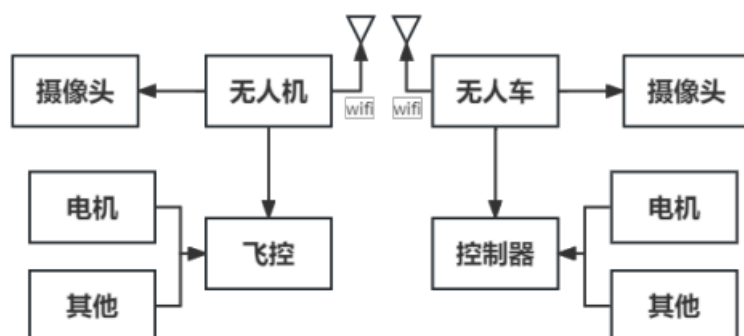
## 10.2 Mediapipe手势识别

MediaPipe是用于构建应用ML管道的多模式（例如，视频，音频，任何时间序列数据）的框架。利用MediaPipe，可以将感知管道构建为模块化组件的图形，包括例如推理模型（例如，TensorFlow，TFLite）和媒体处理功能。



## 四、空地协同控制

### 1. 空地协同通信



无人机与无人车共同连接一个热点，依靠wifi在同一个局域网中实现TCP/IP通信，互相发送ROS话题，无人车通过接收无人机发送的位置话题进行路径跟踪。

## 2. 坐标变换

无人车在地面移动没有高度变化，要想实现空地协同，需将无人机和无人车处在统一坐标系下，并保证无人车的位姿变换中 $z$ 始终保持常值。

# 五、调试过程中遇到的问题

## 1. 定位毯坐标反转

在使用定位毯进行Stanley路径跟踪有时会出现无人机坐标反转的情况。经过我们反复的实验发现，当无人机从初始状态开始在定位毯上进行路径跟踪时并不会出现坐标反转的情况，而在进行二维码跟踪后进入定位毯后一定会出现坐标反转，所以我们推断导致该情况出现的原因是二维码所在坐标系和定位毯所在坐标系出现不一致现象。

解决方案：使用老师提供的坐标反转节点将无人机坐标反转回去。

## 2. 无人机起飞不稳

在无人机起飞时会出现左右晃的状态导致无人机imu出现混乱，无法正常飞行。经过我们实验后发现，在无人机飞行前开启二维码跟踪节点会导致起飞过程中无人机同时接收到上升和朝二维码飞行的两个速度话题，从而导致无人机状态不稳。

解决方案：在无人机起飞时先将二维码跟踪节点关闭。

## 3. PID调试

PID调试过程中出现过度抖动，尤其是在圆环穿越的时候，无人机PID中微分环节系数过大导致的震荡幅度超过所要调整的移动位置，使得无人机无法正常通过。

解决方案：适度调节PID，边调节边实验，适度降低穿越圆环的需要达到的条件。

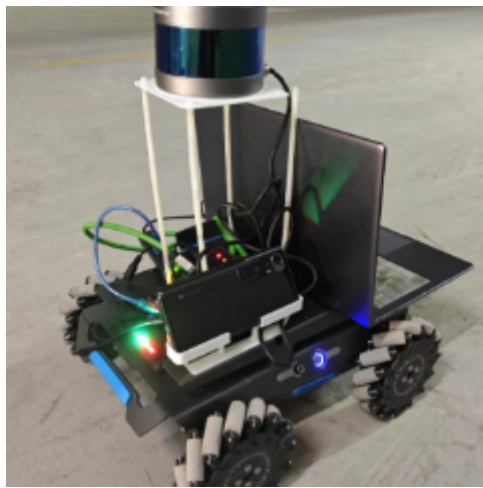
## 4. 无人机无法起飞

在无人机第二次起飞时，往往会出现无法起飞的情况，经过我们实验后发现，在无人机起飞时开启smach状态机后无法起飞，所以我们推断无人机处理节点数过多出现自锁现象。

解决方案：在无人机起飞时先将状态机关闭，等待无人机稳定之后再开状态机。

# 六、总结

经过本次课程我学到了很多关于无人控制方面的知识，掌握了如何使用ROS操作系统完成简单的空地无人协同任务，对以后在机器人操作方面的发展具有很好的启发作用。在学习本课程之前我也做过一些其他关于ROS的项目，比如使用ROS驱动麦克纳姆轮实现自主巡检任务，以下是该项目的实物图片。



该项目主要通过三维激光雷达进行SLAM建图任务，利用RGB-D相机进行墙体裂缝检测。在学习本课程之前我对于该项目的路径规划部分还缺乏一定的了解，通过本课程老师的讲解，我逐步确立了研究的方向，并成功将Bezier曲线路径平滑算法和Stanley路径跟踪算法应用到了该项目中，最终获得了省级大创优秀结题。

本次课程让我明白了理论和实践具有一定的区别，让我学会了如何在实践过程中随机应变，解决问题。在该课程的调试过程中，我们遇到了很多琐碎的问题，比如无人机定位不准、无人车雷达数据漂移等，这些问题在理论上并没有给出确定的解决方法，我们只有自己进行实验判断，不断试错，找出规律，找到解决方法，其实我觉得这正是实践的目的。

有了本次课程的经验，我对ROS有了更深层次的了解，也越发对ROS的一些实践项目感兴趣，于是我参加到了今年一个关于ROS无人机的大创项目，并作为主要负责人，准备继续对ROS进行进一步的学习和研究。

在本次课程的学习中，不论是老师还是助教都十分热情，很有耐心地解决我们提出的问题，这也让我能够从他们那里学到更多的知识。每位老师的课堂都给我带来了许多收获，我看到了机器人技术的发展历程，了解到了机器人各个前沿领域的研究，这些知识让我开始对自己以后研究方向有了想法。