自己动手做一台 SLAM 导航机器人

附录 C：如何选择 ROS 机器人平台进行 SLAM 导航入门

作者：

知乎@小虎哥哥爱学习

目录

1. SLAM 与 ROS 的关系

1.1.关于 SLAM

在了解 SLAM 之前，需要先对机器人有一个整体的认识。机器人是一个复杂的装置，涉及到执行机构、感知、决策等主要环节。机器人上的配备的常用执行机构有轮式运动底盘、机械手臂、音响和显示屏；机器人上的感知设备通常有激光雷达、声呐、摄像头、IMU、轮式里程计编码盘、麦克风、触摸感应；机器人的决策是机器人智能的体现，机器人通常借助感知装置持续跟外部环境进行交互，从而来获取机器人的状态和环境的状态，我们可以简单的把机器人获取自身状态的行为叫做自我认知，把机器人获取环境状态的行为叫做环境认知。机器人的自我认知和环境认知往往是相辅相成互相作用的，所以这里就不做区分了。由于目前的机器人智能还比较低级，所以这里讲到的机器人认知也是低级别的，例如人脸识别、语音识别、机器人定位、环境障碍物探测。有了认知，机器人就可以帮人类完成很多工作了，例如搬运货物、照看小孩、陪伴闲聊、帮忙管理家里的智能设备、查询天气交通新闻资讯等等。我们可以把机器人帮助人类完成的这些个工作叫做机器人的技能，机器人拥有的这些个技能我们可以简单的理解为机器人低级别的思想。机器人的躯壳+机器人的认知+机器人的思想，基本上就是机器人该有的模样了。

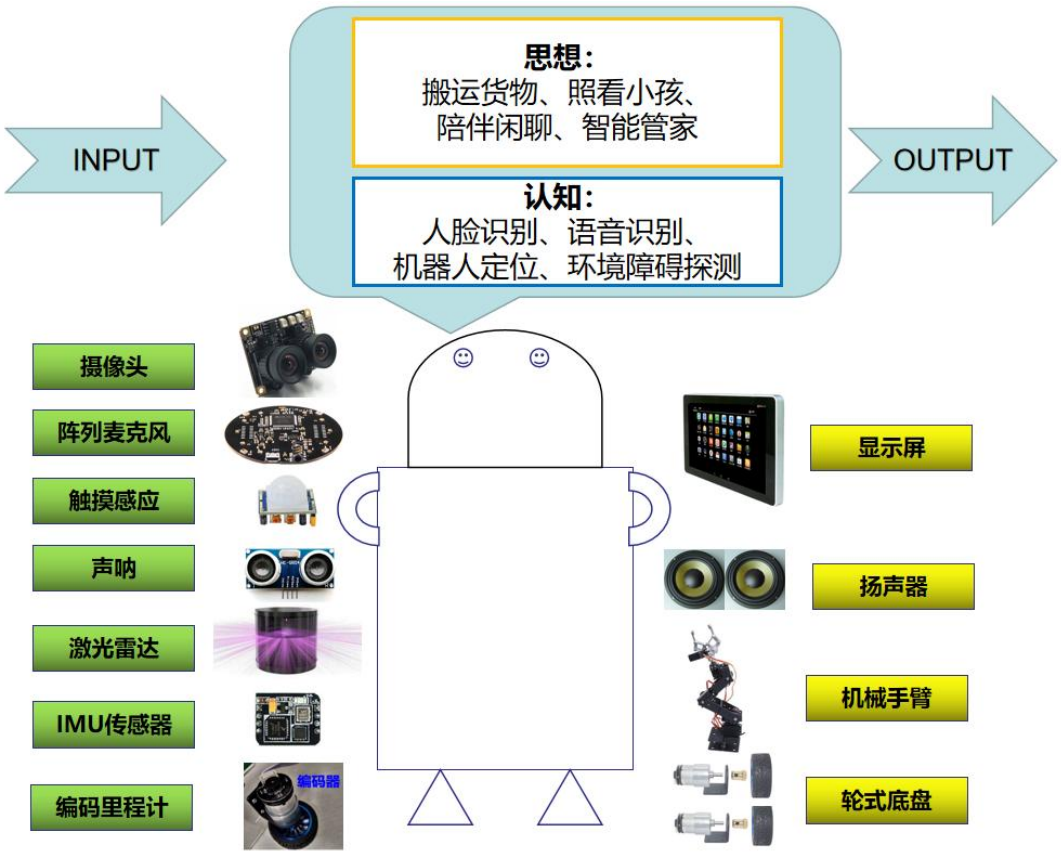

图 1.1.1 一个典型机器人的模样

这样一看，机器人是一个超级复杂的东西，但是机器人上现在关键性的技术就那么几个。其实，学界和工业界热门的研究和开发也是围绕这几个关键技术展开的。关键技术 1：机器人移动底盘，绝大部分机器人需要依靠移动底盘来行走，当然除了水下机器人和腿型机器人这些特殊形态外，电机的效率功率、电机控制电路、编码器检测、底盘运动模型等组成部分，

机器人底盘是 AGV 智能车、家庭服务机器人、农业工业自动化移动平台必不可少的关键组件。关键技术 2：机械手臂，机器人抓取物品、组装零件、写字画画都需要用到机械手臂，高性能的舵机、精密的机械关节、智能关节控制算法等几个组成部分，机械手臂在工业自动化领域已经得到广泛应用，比如汽车自动装配、手机零配件装配，不过应用在服务机器人和 AGV 智能车上的轻便微型机械手臂目前还不成熟或者说还不普及。关键技术 3：SLAM 导航，这里终于讲到了大名鼎鼎的 SLAM 技术了，学术上的 SLAM 是指同时定位和建图技术，但是平时大家说的 SLAM 通常是广义的机器人定位+环境建图+自主导航+动态避障，我们这里也讨论的是广义的 SLAM，简单点说 SLAM 技术帮助机器人认识环境并在环境中自由行走到目的地。关键技术 4：语音识别，语音交互是大部分机器人的标配，人类通过语音交互向机器人下达任务指令，语音交互包括语音识别、语音合成、自然语言处理等部分组成，目前基于语音交互做出来的小机器人也是遍地开花。关键技术 5：物体识别，机器人借助摄像头对自己所处环境的各种事物进行识别，比如流行的人脸识别、车牌识别、手势识别等等。

关键技术 1：机器人移动底盘

关键技术 2：机械手臂

关键技术 3：SLAM 导航

关键技术 4：语音交互

关键技术 5：物体识别

本人在这里总结归纳了机器人上的 5 大关键技术，由于个人能力和篇幅限制原因，接下来的内容将主要涉足机器人 SLAM 导航领域的相关技术。

SLAM 导航技术，主要用来解决机器人的定位、环境建图、自主导航、动态避障等问题。SLAM 这项技术其实已经有几十年的历史了，SLAM 最早是出现在军事应用中，比如勇气号火星探测车，在不能实时遥控的未知环境行星上的探测车为了执行任务，需要借助 SLAM 技术来导航和避障。后来慢慢的 SLAM 技术就从军用转民用了，有了我们现在看到的小到家里的扫地机器人大到无人驾驶汽车的各种 SLAM 应用，还有各种 AR 和 VR 应用很多也用到了 SLAM 技术。

其实单独的定位技术我们日常生活中能见到很多，比如最常用的 GPS 定位技术、wifi 定位技术、磁条导轨定位技术。单独的环境建图技术也有很多成熟的应用，比如医学中的 CT 对人体的全方位扫描技术、电影制作中对某个三维物体的扫描建模、隧道勘探测绘等。但是面对机器人这样一个复杂的应用，单独的定位技术和单独的环境建图技术都不能很好的解决问题，于是结合了定位与建图的 SLAM 技术就出现了。接下来，简单的梳理一下定位、建图、SLAM 同时定位于建图、SLAM 导航技术的理论发展过程。

（1）机器人中的不确定性

机器人所处的环境存在大量不可预测性，机器人中传感器的测量存在噪声和干扰，电机等执行机构也存在一定程度的不可靠性，还有一些不确定性是由机器人的软件导致的。要研究机器人的感知和行为就必须对这些不确定性进行建模，利用概率理论可以明确的表示这种不确定性，并且可以利用概率算法来对概率分布进行推理计算，以数学的方式合理的表示机器人的模糊性和置信度。简单点说，就是通过对机器人的概率建模，可以对机器人的不确定性进行明确的可计算性的表示。

（2）机器人中的状态估计

处理机器人中的这种不确定性的概率技术我们称之为概率机器人技术，其核心是用传感器数据来估计状态的思路。

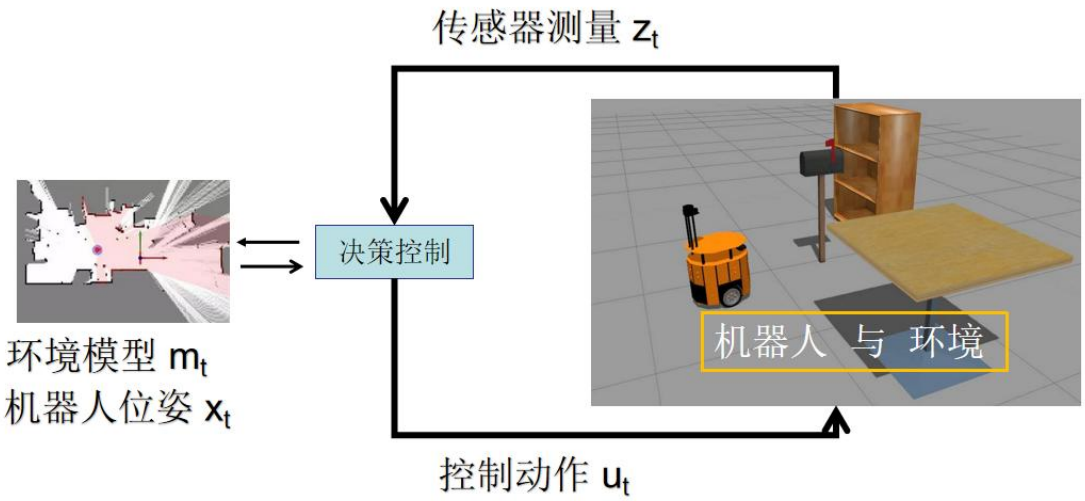

图 1.1.2 机器人与环境交互

机器人通过传感器对机器人自身状态和环境状态进行测量，这里的机器人状态包括机器人位姿、运动里程计信息、线速度和角速度信息等，而环境状态包括环境中物体的位置、物体的特征等。机器人同时利用控制动作来调用执行机构和环境进行交互，这里的控制动作包括机器人运动、物体操纵等。这样，机器人在通过传感器测量和控制动作与环境进行交互的过程中，对环境模型和机器人位姿进行估计。对环境模型的迭代估计就是我们常说的环境建图，对机器人位姿的迭代估计就是我们常说的机器人定位。

利用传感器测量和控制动作对环境模型和机器人位姿进行估计由概率法则迭代计算来完成。控制动作的概率模型我们称之为运动概率模型，传感器测量的概率模型我们称之为测量概率模型，这里的概率法则迭代计算过程就是大名鼎鼎的贝叶斯迭代网络。

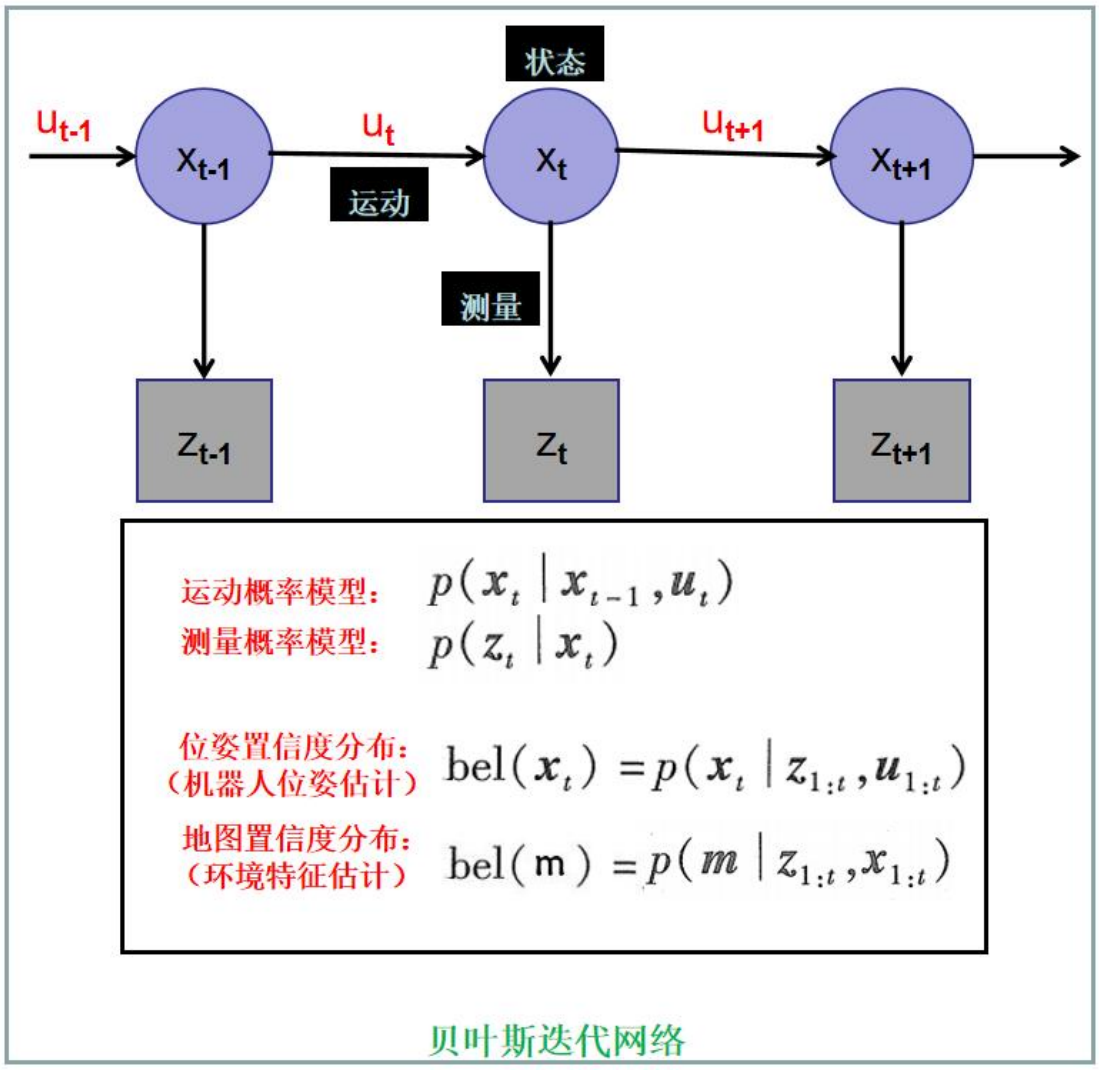

图 1.1.3 贝叶斯迭代网络

（3）贝叶斯滤波及其各种实现算法

通过对机器人中状态估计的了解，我们知道了贝叶斯概率法则起着重要的作用，而且大部分计算置信度的通用算法都是由贝叶斯算法给出的。这里说的贝叶斯算法也称作贝叶斯滤波，该算法根据测量和控制数据计算置信度分布 bel()。贝叶斯滤波是一种递归算法，也就是说 t 时刻的置信度由 t-1 时刻的置信度来计算。

贝叶斯滤波算法具有两个基本步骤，即预测和更新。根据运动概率模型，可以对初步的预测置信度分布；然后利用测量概率模型，对上一步预测得到的置信度分布做进一步的更新计算。贝叶斯滤波的具体实现算法有多种，根据置信度表示方式的不同，贝叶斯滤波的具体实现算法可以分为参数化实现算法和非参数化实现算法两种。参数化实现算法，就是将置信度分布用多元高斯分布的参数进行表示。根据采用不同高斯分布的参数，又可以分为卡尔曼滤波和信息滤波。非参数化实现算法，就是将置信度分布用有限数量的具体值来近似表示。根据采用不同具体值近似的方法，又可以分为直方图滤波和粒子滤波。

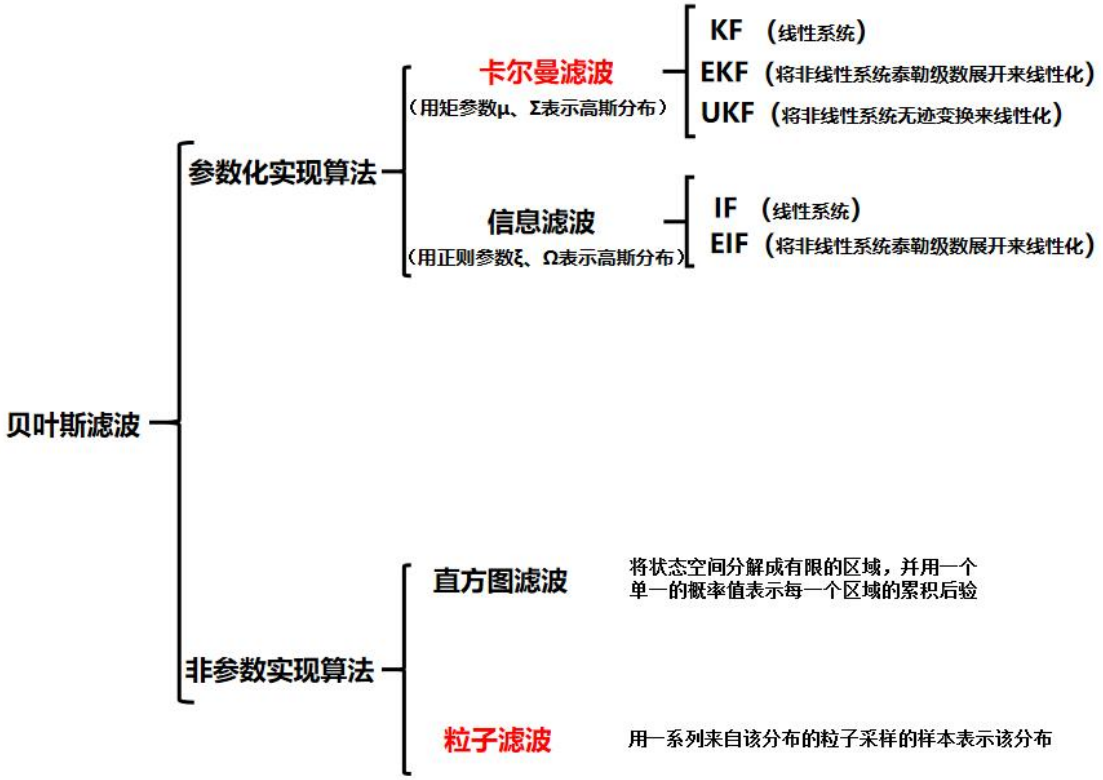

图 1.1.4 贝叶斯滤波及各种具体实现算法

由于贝叶斯滤波理论及各种具体实现算法涉及到大量深奥的数学知识，由于篇幅和个人能力受限，就不展开了，有兴趣的朋友可以参阅《概率机器人》这本书，这里面有细致的讲解和推导。为了加深大家在阅读和编写代码时的理解，这里贴出了贝叶斯滤波的伪代码实现，和机器人中广泛应用的卡尔曼滤波和粒子滤波的展示实例。

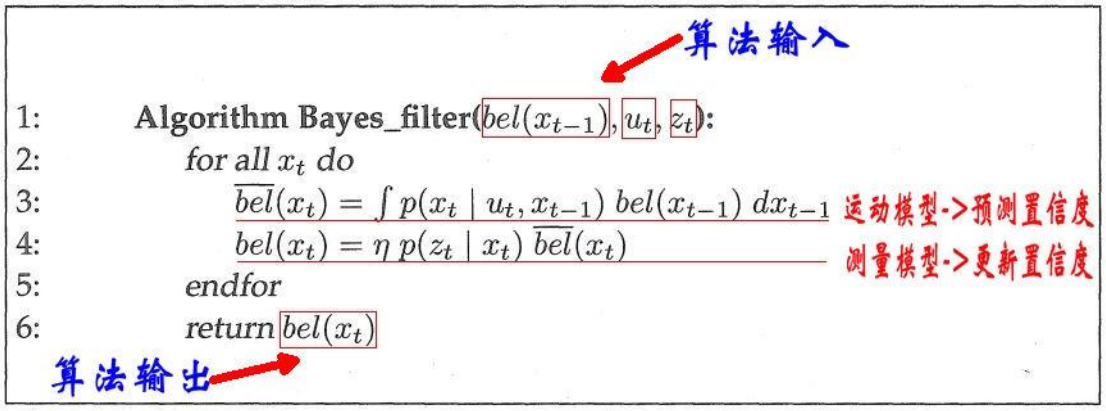

图 1.1.5 基础贝叶斯滤波的伪代码实现

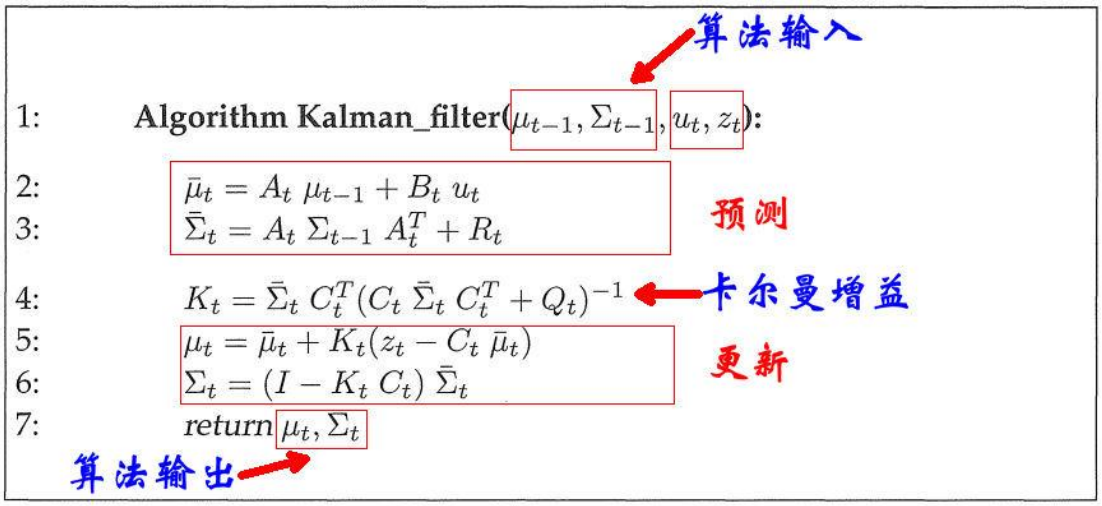

图 1.1.6 卡尔曼滤波的伪代码实现

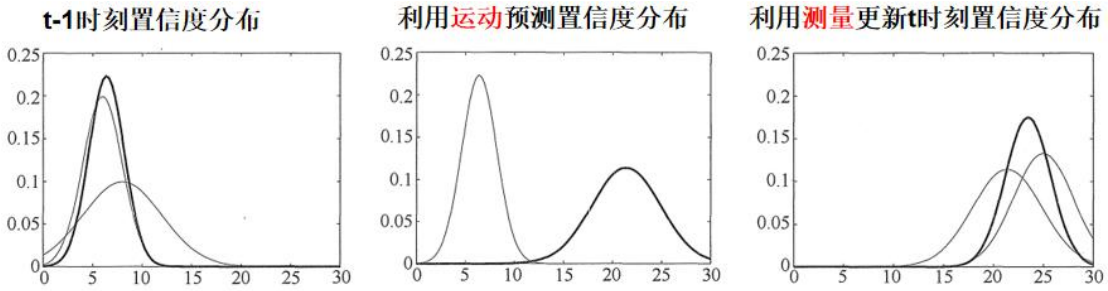

图 1.1.7 卡尔曼滤波对置信度分布的更新过程

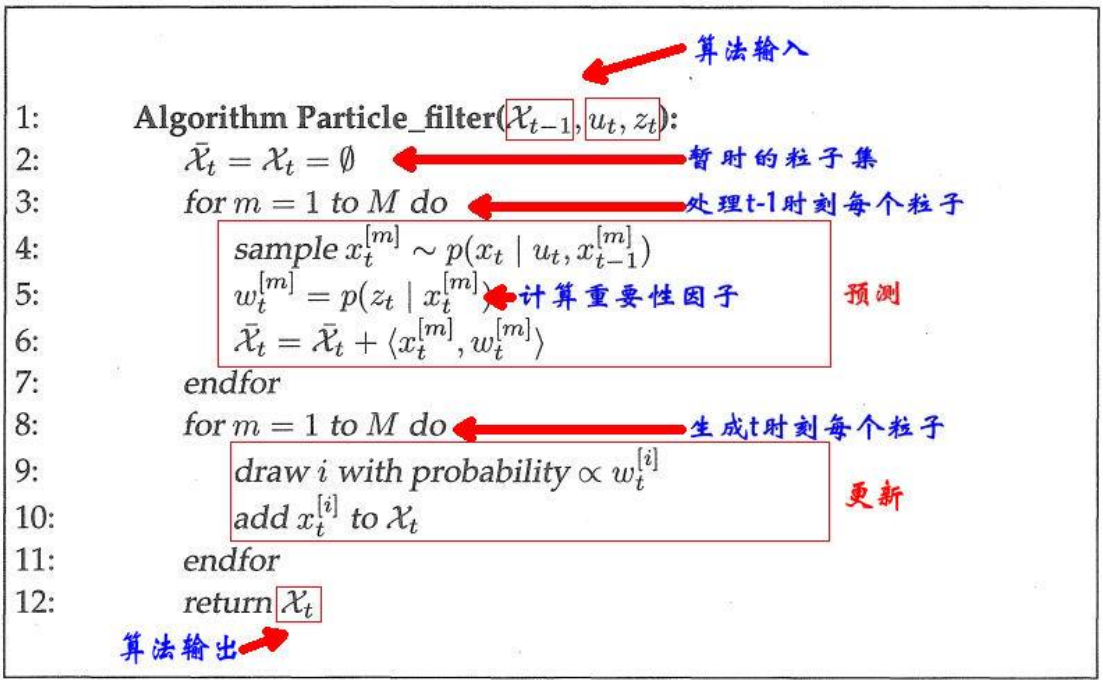

图 1.1.8 粒子滤波的伪代码实现

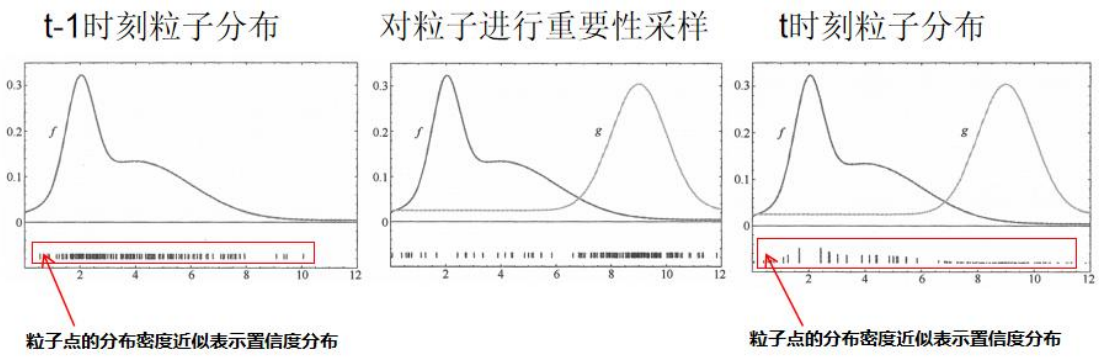

图 1.1.9 粒子滤波对置信度分布的更新过程

（4）机器人中的运动与测量概率模型

了解了机器人中的状态估计及各种滤波概率算法后，我们发现机器人的运动和测量概率模型对算法中的预测和更新步骤起着至关重要的作用。

运动模型我们都知道，就是机器人一个地方运动到另一个地方，位姿变化与运动控制量之间的关系。那什么叫运动概率模型呢？由于控制量作用在机器人上使机器人的位姿变化的过程是一个存在误差的不确定过程，所以控制量作用后产生的结果需要用一个概率分布来描述，这就是运动概率模型。根据运动控制量的不同，可以分成速度运动模型和里程计运动模型。速度运动模型中，控制量是线速度和角速度；里程计运动模型中，控制量是机器人的前后时刻的相对位姿。速度运动模型中，线速度和角速度会受到机器人底盘构造的约束。实验表明，里程计运动模型虽然仍存在误差，但比速度运动模型要更精确。

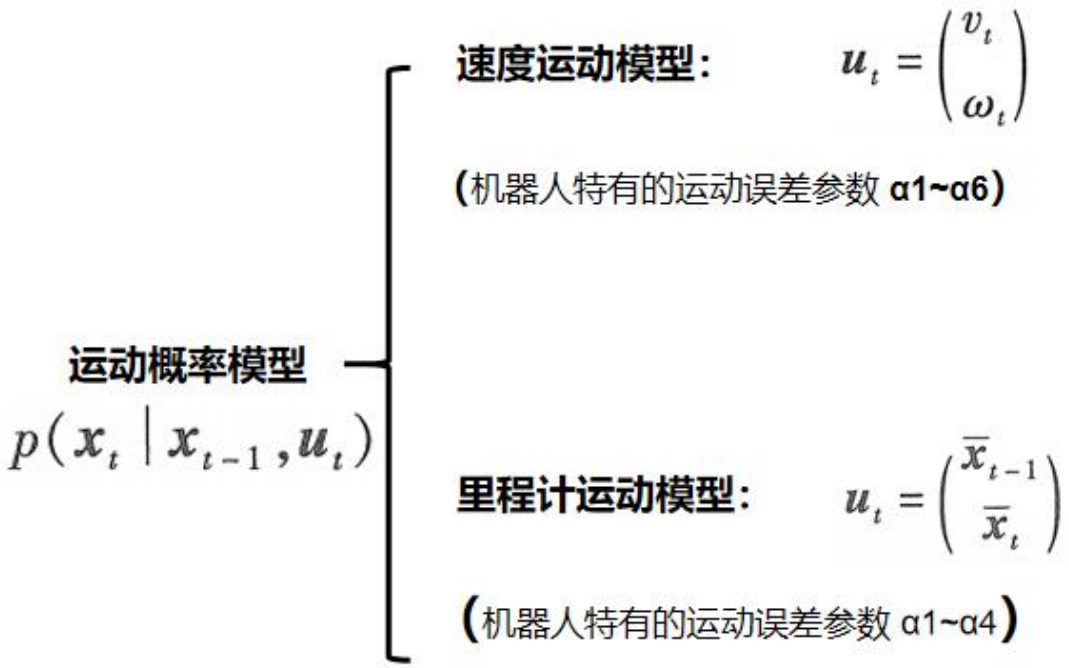

图 1.1.10 机器人的运动概率模型

同样的道理，机器人上的传感器在测量过程中存在误差，所以测量结果需要用一个概率分布来描述。主要讨论当下最流行的机器人测距仪，即激光扫描测距仪。由测距仪的近似物理模型，可以沿着一个波束测量到附近物体的距离，也就是波束模型。另一种测距仪模型不依赖任何传感器的物理模型，而是将传感器扫描到的终点值映射到地图的全局坐标空间，这就是似然域模型。估计一个测量和地图之间的相关性中，测距仪用到了地图匹配模型。还有

一种是从传感器原始测量中提取特征，提取出来的特征我们可以称之为路标，这就是路标模型。

图 1.1.11 机器人的测量概率模型

（5）移动机器人定位与建图

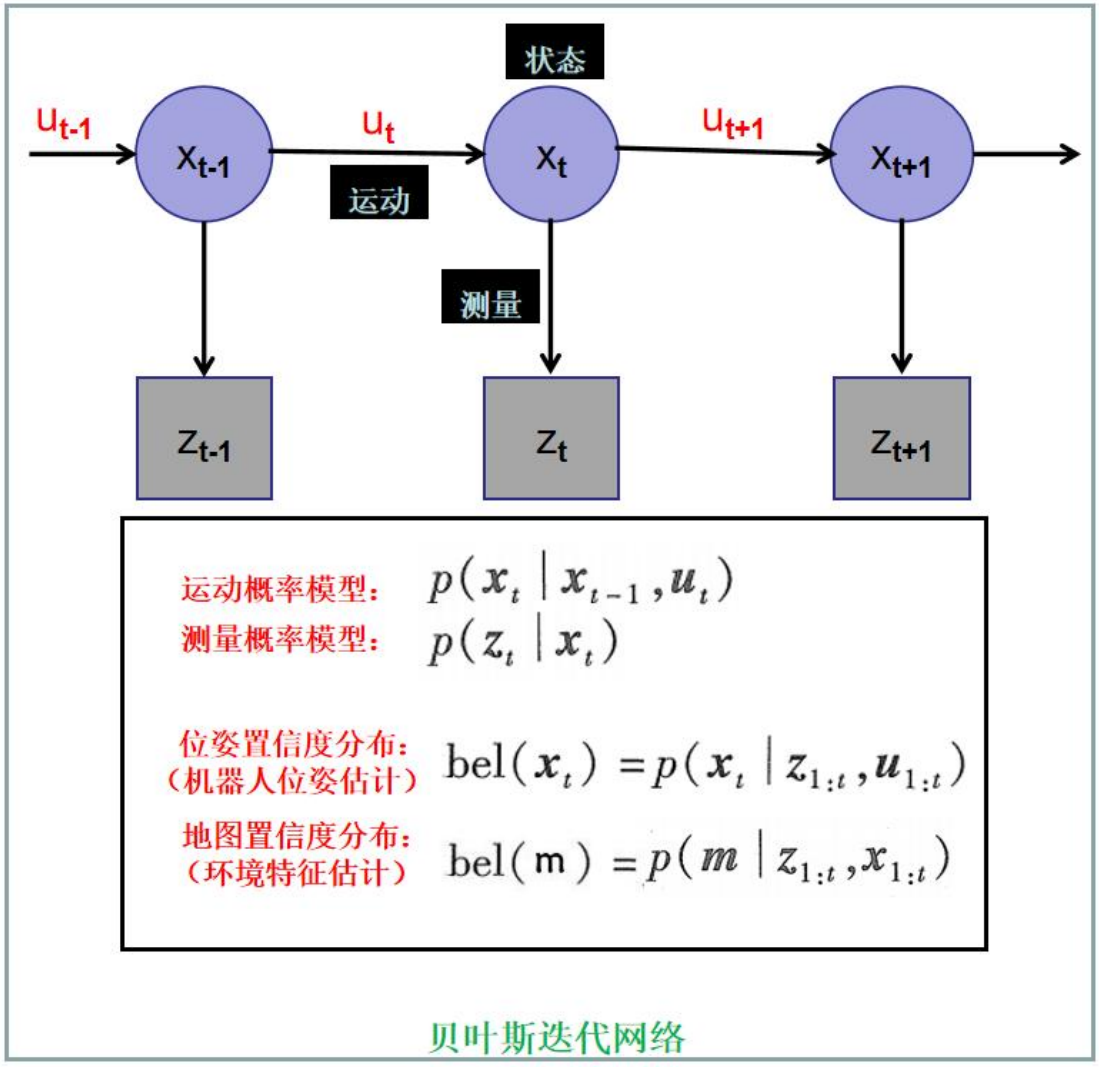

图 1.1.12 移动机器人定位与建图

在了解了机器人中的状态估计、状态估计的各种概率实现算法、机器人运动与测量概率模型，我们再来重新理解上面这样一个经典的贝叶斯迭代网络，就很好理解了，这个里面包

含了机器人定位和建图两大问题。机器人的定位其实就是位姿估计问题，位姿估计的概率实现算法可以有上面提到的 EKF、AMCL 等算法来实现，由于这两种算法在机器人位姿估计中都很流行，所以就不多展开了。机器人的建图其实就是环境特征估计问题，同样也可以用概率的方法就行估计。

（6）SLAM 同时定位与建图

上面已经单独的提出了机器人定位与机器人建图的方法，但是独立的定位问题是建立在地图已知的情况下的，单独的建图问题也是建立在定位已知的情况下的。当机器人不能得到环境地图，也不知道自身位姿的时候，SLAM 问题就出现了。也就是说 SLAM 要同时的进行机器人定位和建图，这个问题比单独的定位和单独的建图都要难得多。

在讨论具体的 SLAM 算法之前，我们先要了解 SLAM 的两种主要形式：在线 SLAM 问题和全 SLAM 问题。

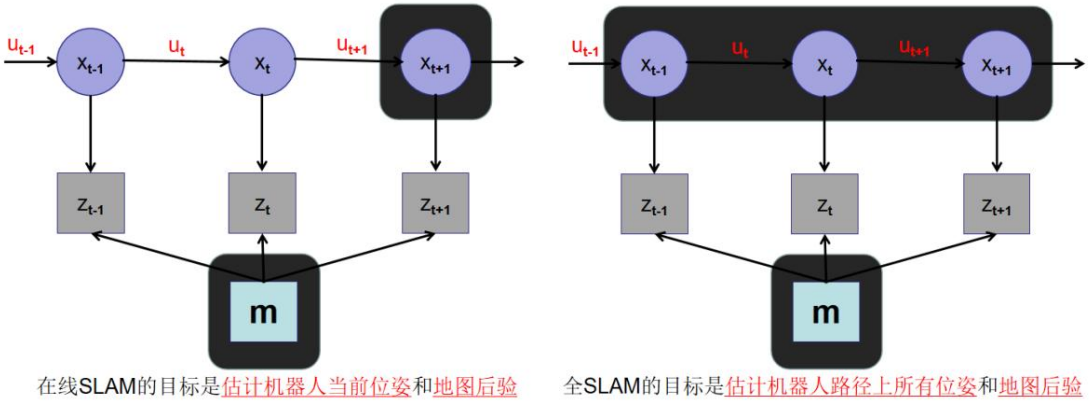

图 1.1.13 在线 SLAM 问题和全 SLAM 问题

在线 SLAM 算法的代表是 EKF SLAM，历史上最早并可能是最有影响力的 SLAM 算法，可以说是 SLAM 研究的元老级算法。EKF SLAM 中的地图是基于特征的，地图由点地标组成。除了估计机器人当前的位姿，EKF SLAM 算法还估计路径上遇到的所有地标的坐标，也就是机器人位姿和地图地标点包含进联合状态矢量里，算法对该联合状态矢量进行估计。

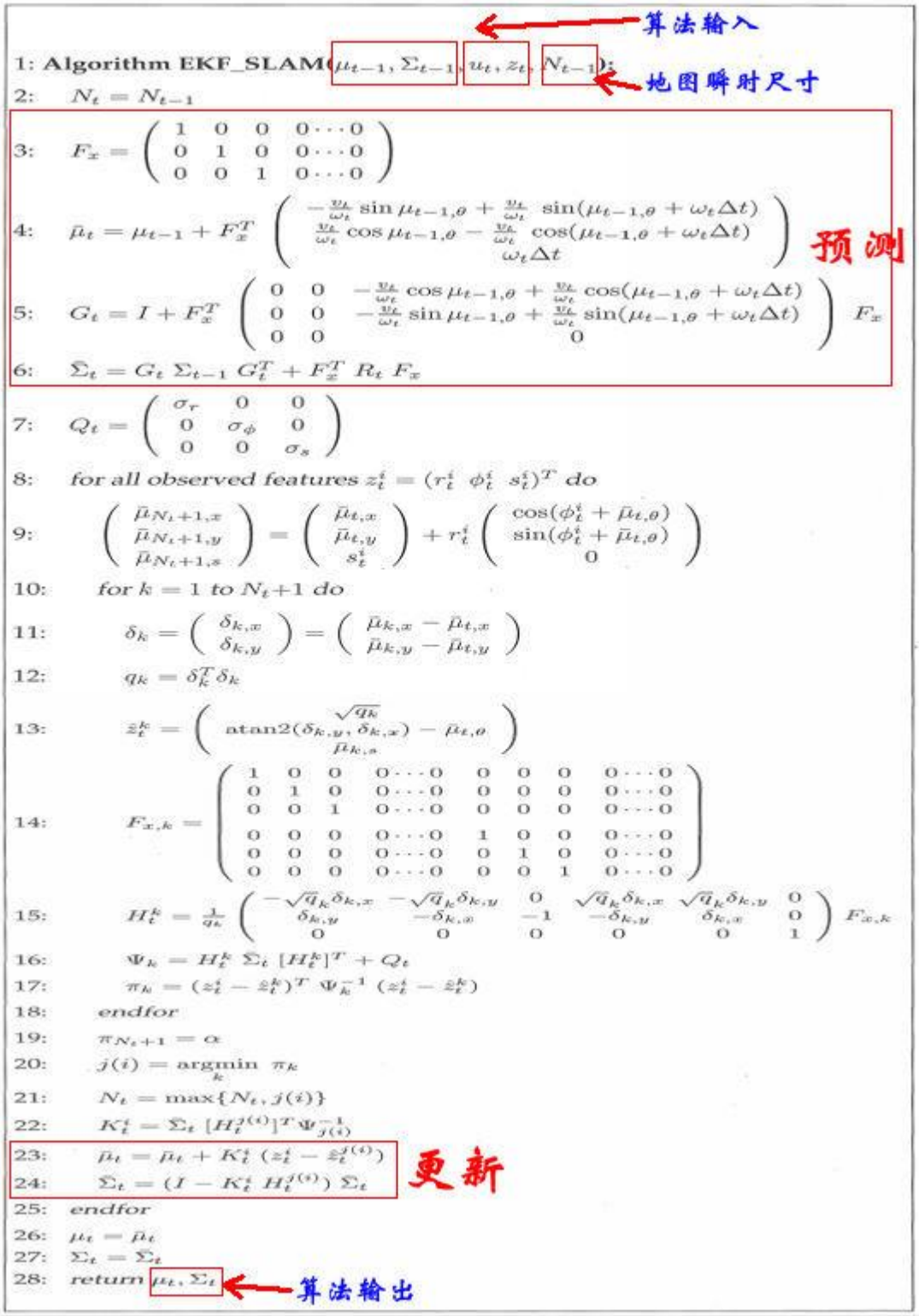

图 1.1.14 EKF SLAM 伪代码实现

全 SLAM 算法的代表是 GraphSLAM，该算法将机器人的运动路径和测量组件成一个软约束的图，利用图论中的优化算法对整个图中的轨迹点和测量点进行估计。由于图的稀疏性特点，可以大大加快计算。

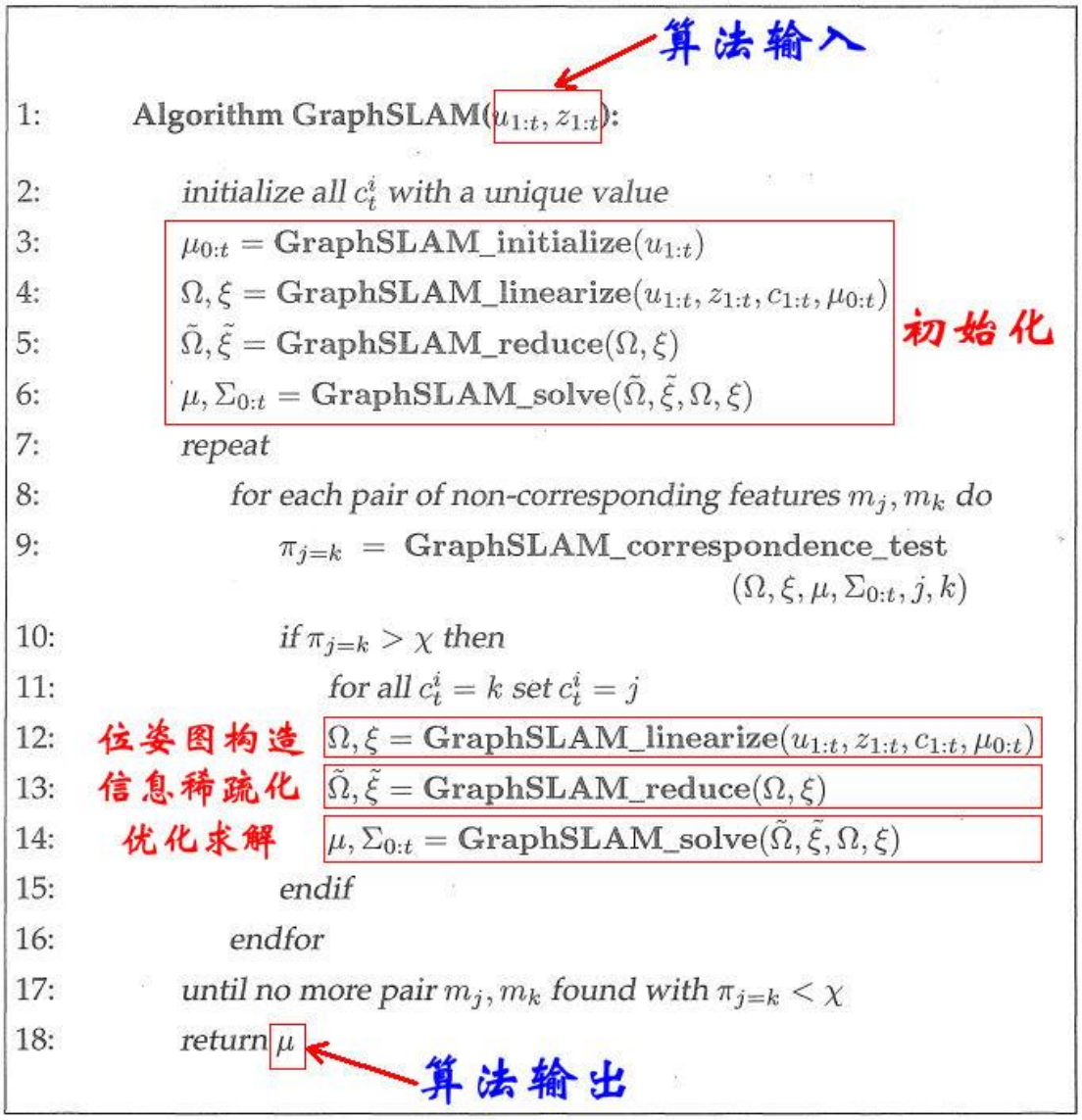

图 1.1.15 GraphSLAM 伪代码实现

EKF SLAM 和 GraphSLAM 是两个极端。EKF SLAM 需要取得每一时刻的信息，把信息分解为概率分布，因此每一步的计算代价都非常昂贵。而 GraphSLAM 刚好相反，只是简单的积累每一时刻的信息，也就是简单的将收到的信息存储下来，然后可以离线的进行推理的步骤，加之存储下来的信息的稀疏性，因此 GraphSLAM 计算的开销是比较小的，但是随着地图规模扩大算法会消耗越来越多的内存直至崩溃。面对这两个极端问题，当然就会有介于 EKF SLAM 和 GraphSLAM 是两个极端之间的折中的方法，就是 SEIF SLAM。SEIF SLAM 算法继承了 EKF SLAM 信息表示的高效性，也保留了 GraphSLAM 计算代价小的优点，可以说 SEIF SLAM 是高效和可实现的 SLAM 算法。还有另外一种高效和可实现的 SLAM 算法，就是 FastSLAM，该算法使用粒子滤波估计机器人的路径，我们都知道粒子滤波和众多基于参数化的滤波算法相比存在计算开销小和便于处理非线性模型的优势，基于 FastSLAM 的多个变种算法在机器人已经得到广泛的应用了，比如 gmapping 等等，由于篇幅限制就不展开了。

（7）现今主流的 SLAM 算法

现今在机器人上使用最广泛的应该算激光 SLAM 了，在扫地机器人、服务机器人、AGV 智能车上普遍搭载了单线激光雷达 SLAM 算法，像无人驾驶汽车、户外机器人则普遍搭载了多线激光雷达 SLAM。另一种热门的研究是视觉 SLAM，视觉 SLAM 有配备单目、双目、深度

相机的多种形态，并且根据采用视觉特征点的区别还有直接法、半直接法、稀疏法之分。然后还有就是各种复合式的SLAM算法，比如激光与视觉融合的SLAM、融合了IMU的视觉SLAM。最后，就是一些最新颖的 SLAM 算法，比如用深度学习来做的端到端的 SLAM、基于物体识别的语义 SLAM。由于本文的重点不是 SLAM 综述，所以具体的算法性能比较就不展开了，有兴趣的朋友可以参阅相关 SLAM 综述文章。

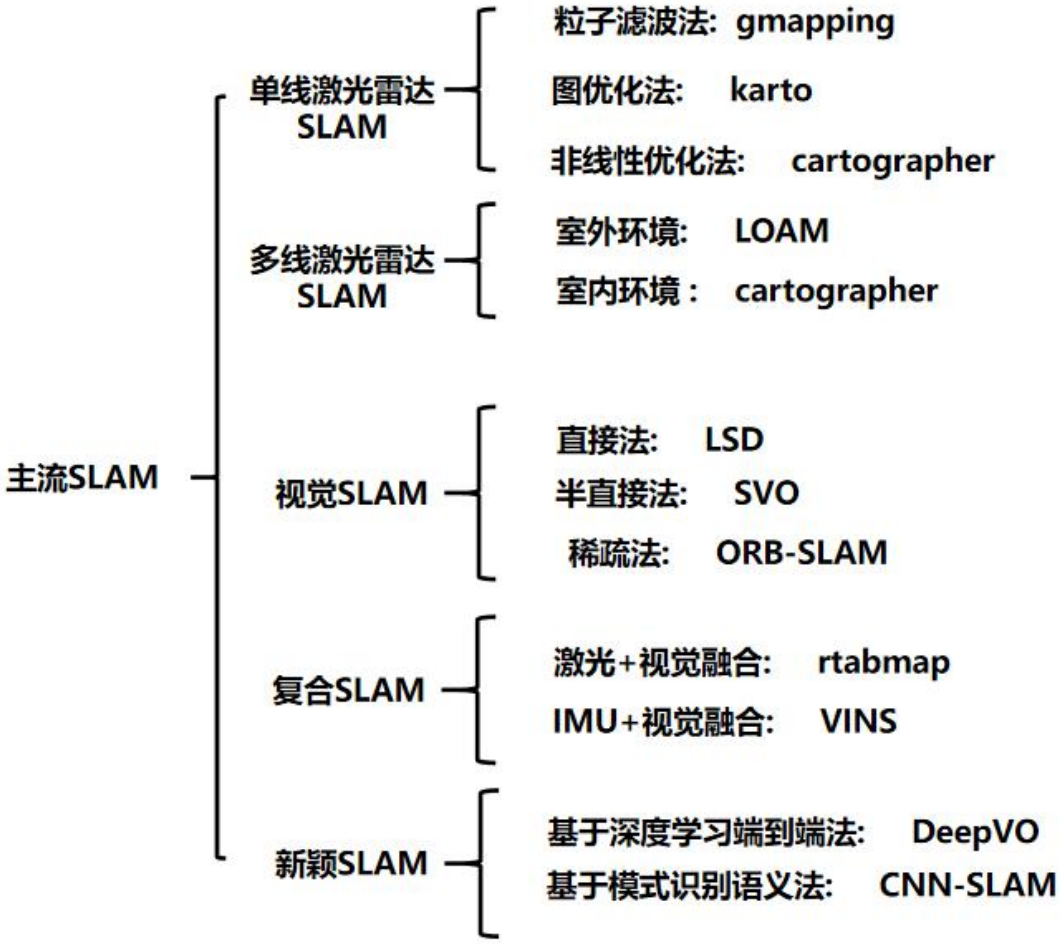

图 1.1.16 现今主流的 SLAM 算法

（8）机器人自主导航与动态避障

机器人用 SLAM 构建出了环境的地图，在已知了环境地图的情况下，可以用 SLAM 的重定位功能或者单独的基于已知地图的定位算法比如 AMCL 来进行机器人的定位。环境地图和机器人位姿都有了，就可以开始来做自主导航和避障了。机器人自主导航可以分成两个实现部分，第一个部分就是路径规划，第二个部分就是控制策略。路径规划利用地图信息寻找一条能到达目标的全局路径，全局路径在机器人导航过程中起到全局战略性的指导。理想情况是，机器人完全按照全局路径移动到目标，但是实际环境往往是多变和复杂的，而且机器人实际控制也会存在偏差，所以机器人的实际运动控制需要有一套控制策略来最终实现。控制策略需要尽量逼近全局路径、尽量远离障碍物、最快时间到达目标等因素，这些因素可以用一个回报函数来评价，寻找最佳控制策略的过程中递归的计算每一次行动的回报函数值。这样控制策略在回报函数的指引下，就可以给出最佳的控制策略，控制策略控制机器人完成实际的移动。

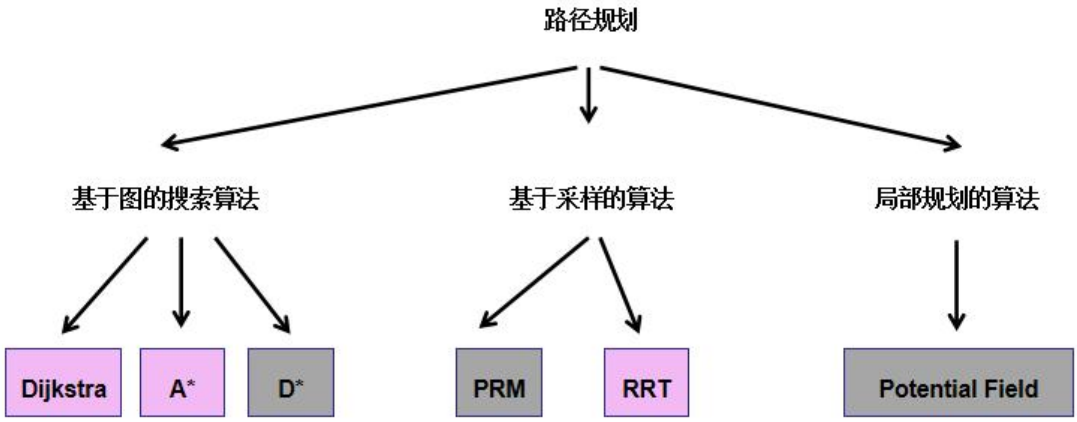

图 1.1.17 主流路径规划算法

1.2.关于 ROS

ROS 机器人操作系统在机器人应用领域很流行，依托代码开源和模块间协作等特性，给机器人开发者带来了很大的方便。

ROS 是一个适用于机器人的开源的元操作系统。其实它并不是一个真正的操作系统，其底层的任务调度、编译、寻址等任务还是由 Linux 操作系统完成，也就是说 ROS 实际上是运行在 Linux 上的次级操作系统。但是 ROS 提供了操作系统应用的各种服务（如：硬件抽象、底层设备控制、常用函数实现、进程间消息传递、软件包管理等），也提供了用于获取、编译、跨平台运行代码的工具和函数。ROS 主要采用松耦合点对点进程网络通信，目前主要还是支持 Ubuntu 系统，windows 和 Mac OS 目前支持的还不好，所以推荐在 Ubuntu 系统上安装使用 ROS。

总结起来就是，使用 ROS 能够方便迅速的搭建机器人原型。ROS 使用了 BSD 许可证，这是一个很宽松的开放许可证，允许在商业和闭源产品中使用，这一点对开发产品的创业公司很重要。ROS 当前的代码统计量，总行数超过 1400 万，作者超过 2477 名。代码语言以 C++为主，63.98%的代码是用 C++编写的，排名第二的是 python，占 13.57%，可以说 ROS 基本上都是使用这两种语言，来实现大部分的功能。

图 1.2.1 ROS 的图标

大家一听到 ROS 机器人操作系统，就被操作系统几个字给吓到了。其实，ROS 就是一个分布式的通信机制，帮助程序进程之间更方便的通信。由于机器人是一个多部件的设备，四肢、视觉、听觉等部位都有配套的程序进行控制，那么要协调这些部位怎么办呢？就需要让这些分散的部位能够互相的通信，这正是 ROS 被设计的最初目的。随着越来越多的人参与到 ROS 的开发完善，ROS 才发展成今天很完善很系统的样子来，涵盖了大量的第三方工具和大量的实用的开源算法软件包。搞懂了 ROS 的通信机制后，机器人的各种算法的开发还是基于我们常见的 C++和 Python 的。

节点是主要的计算执行进程，功能包中创建的每个可执行程序在被启动加载到系统进程

中后，该进程就是一个 ROS 节点，node1、node2、node3 等都是节点（node）。节点都是各自独立的可执行文件，能够通过主题（topic）、服务（server）或参数服务器（parameter server）与其他节点通信。ROS 通过使用节点将代码和功能解耦，提高了系统的容错力和可维护性。所以你最好让每一个节点都具有特定的单一的功能，而不是创建一个包罗万象的大节点。节点如果用 c++进行编写，需要用到 ROS 提供的库 roscpp；节点如果用 python 进行编写，需要用到 ROS 提供的库 rospy。

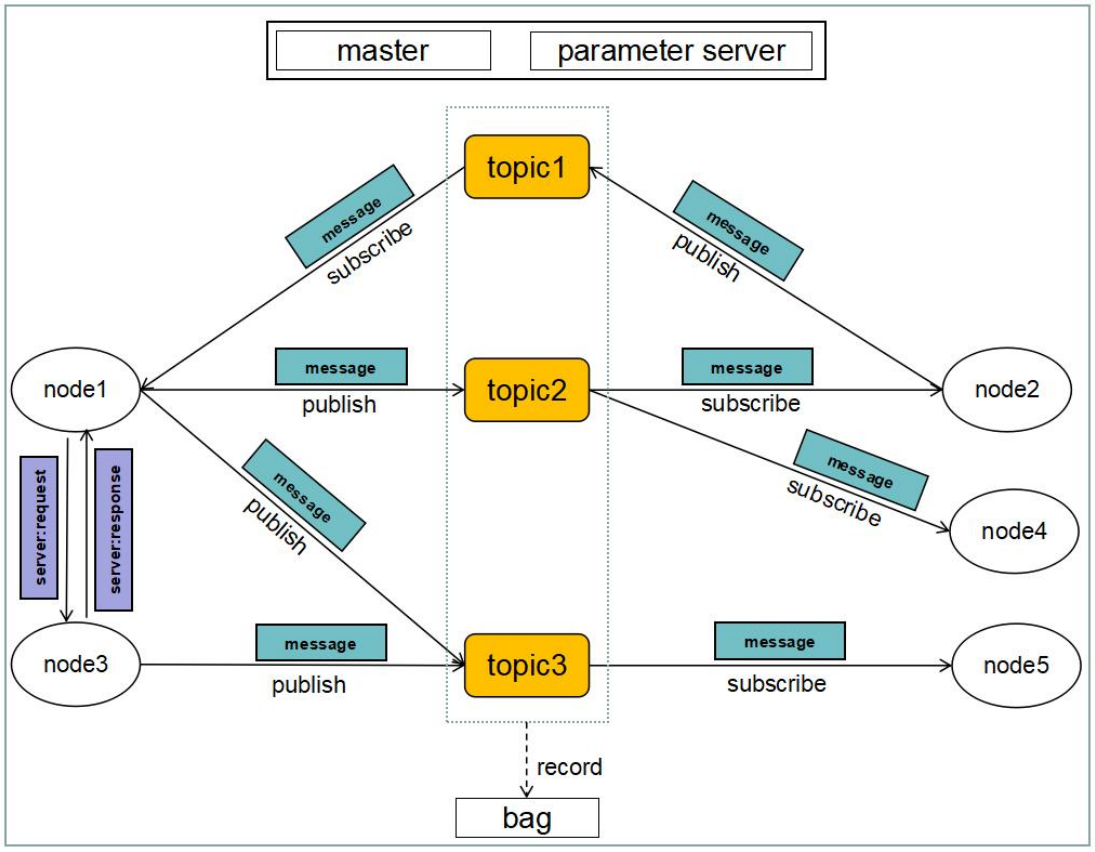

图 1.2.2 ROS 网络通信的架构

如果你是刚刚接手 ROS 方面的开发或项目，你肯定会觉得 ROS 中的各种概念非常奇怪，但是当你对 ROS 的使用熟练之后，你就觉得这些概念很好理解了。与其他操作系统相似，一个 ROS 程序的不同组件要被放在不同的文件夹下，这些文件夹是根据不同的功能来对文件进行组织的。

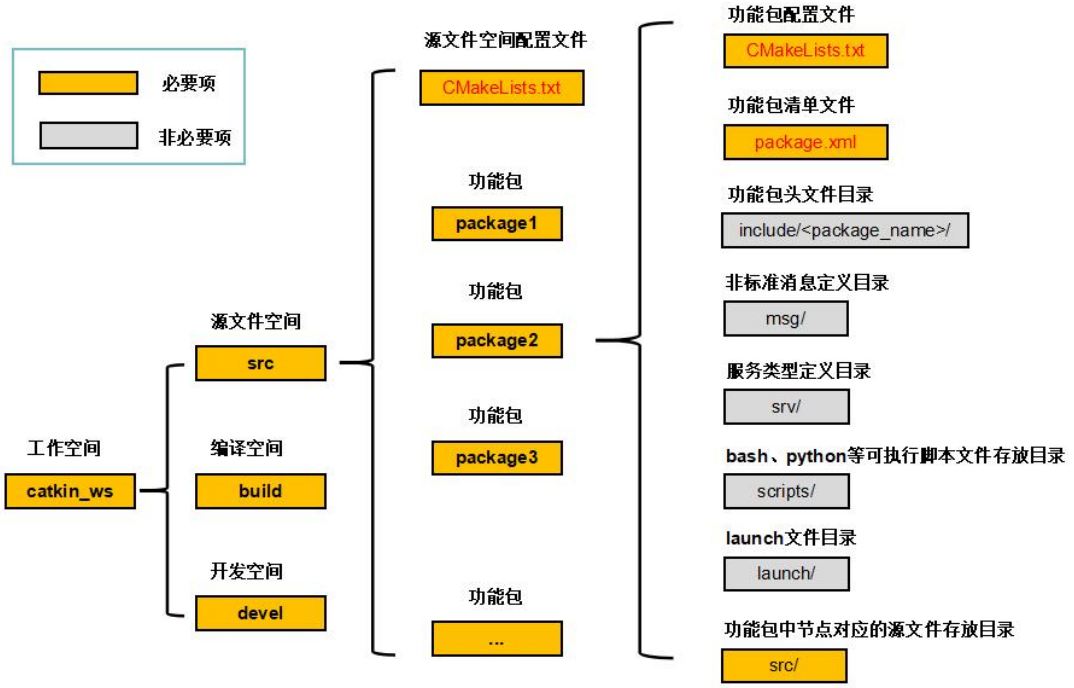

图 1.2.3 ROS 程序的组织形式

1.3.SLAM 与 ROS 的关系

SLAM 最核心的地方在算法，侧重点在于如何构建出效果好的地图，并为机器人导航提供更好的数据保障。ROS 帮忙解决传感器驱动、显示、各种核心算法间的沟通协调问题。如果做商用产品就另当别论了，商业的产品一般会专门开发自己的一套驱动、调度、显示的系统，或者拿 ROS 系统来裁剪以保障稳定性和效率。

2. ROS 移动机器人的整体构造

2.1.硬件构造解析

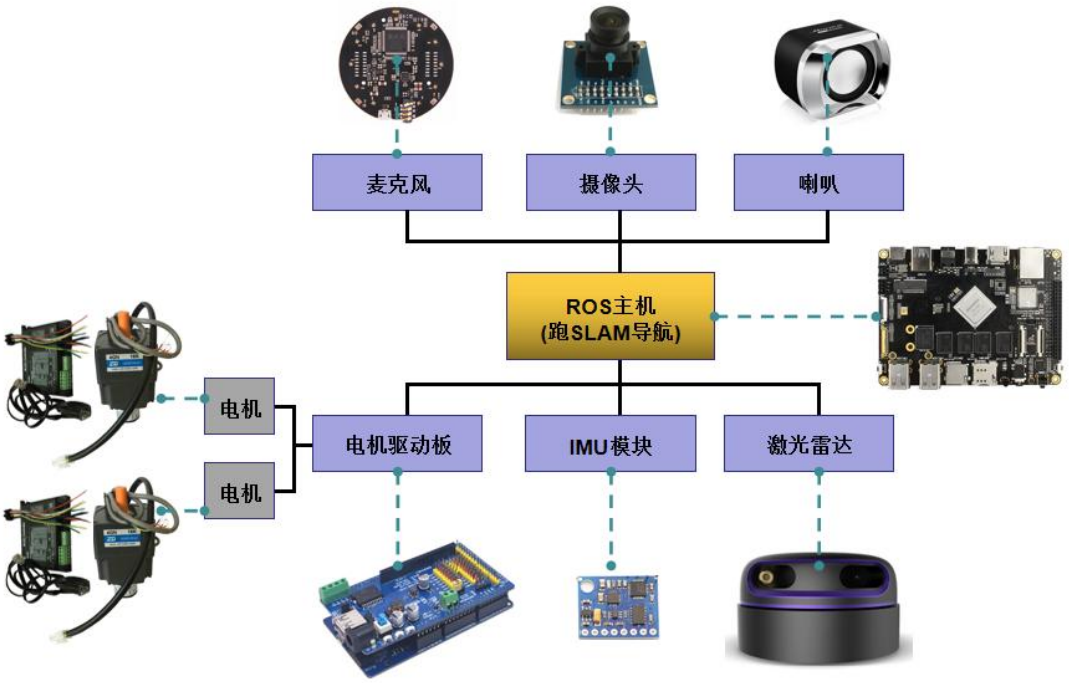

图 2.1.1 一个典型的 ROS 移动机器人的硬件构造

一个典型的 ROS 移动机器人的硬件构造包括：带编码器的减速电机、电机驱动板、IMU 模块、激光雷达、ROS 主机、麦克风、摄像头、喇叭。带编码器的减速电机、电机驱动板、IMU 模块、激光雷达应该是 SLAM 导航避障的标配硬件，ROS 主机是机器人的计算中心（运行 SLAM 导航算法等），麦克风、摄像头、喇叭是完成语音交互和图像感知时需要的。

2.2.电机解析

（1）电机的种类

按照电机的电源类型分为直流电机、交流电机，按照电机的换相方式分为有刷电机、无刷电机，按照电机转子的构造可以分为内转子电机、外转子电机。

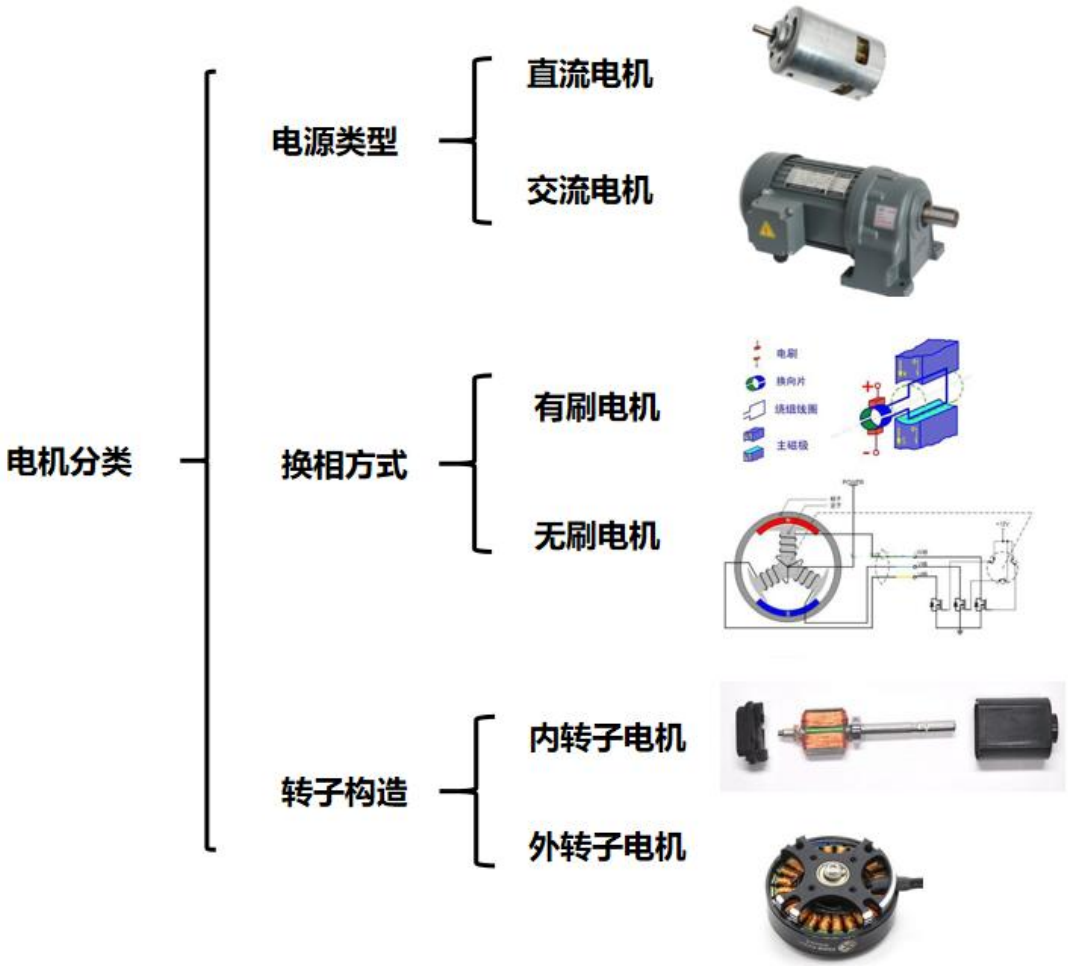

图 2.2.1 电机的种类

（2）减速箱

电机在实际的使用中，往往配合减速箱一起使用，来提高输出力矩和载重性能。减速箱有普通的减速箱和行星减速箱。行星减速箱的优势是产生同等减速比条件下减速箱体积可以更小，一般一些对体积敏感的地方用的多吧。

（3）编码器

编码器起着对电机转速测量反馈的重要部件，有霍尔编码器、光电编码器、碳刷编码器等，编码器通常采用正交编码去形式进行信号输出，也就是 AB 两相信号正交输出，有些还会有 Z 相信号对转过一圈进行脉冲输出。

（4）电调

电调产生电机需要的相位信号，并且通过 PWM 等方式对电机转速进行调节。像一些小功率的直流有刷电机，电调其实就是一个 H 全桥电路，PWM 信号控制 H 桥的四个臂起到控制电机转速和转向的目的。像一些大功率的直流无刷电机，电调的构造就比较复杂了，电调需要实现多相控制信号产生、电子换相、保护等。

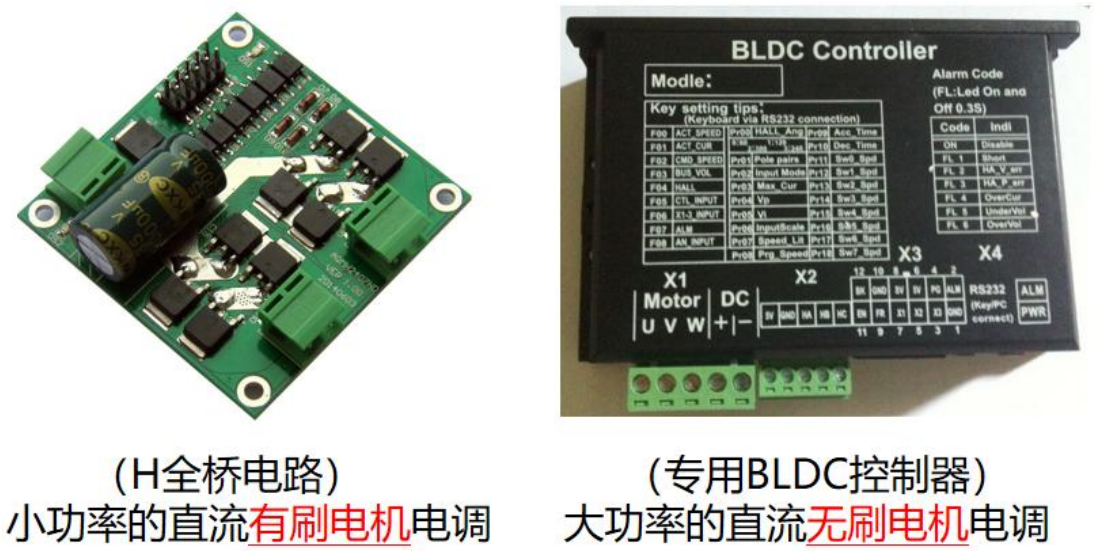

图 2.2.2 电调

（5）移动机器人电机选型

电机是移动机器人的动力核心，所以选择合适的电机对一个移动机器人还是很重要的。电机的选型不仅要考虑参数性能，比如转速、力矩、载重、体积，还要考虑成本、安全、可实现性。如果是 10KG 载重级别的小型移动机器人，可以选择 DC 12V 带编码直流有刷减速电机就可以了。如果载重在 30KG 及以上的移动机器人的话，就需要选择功率和效率更高的无刷电机了，无刷电机的功率和减速箱参数可以根据实际机器人的载重和运动速度来选择。

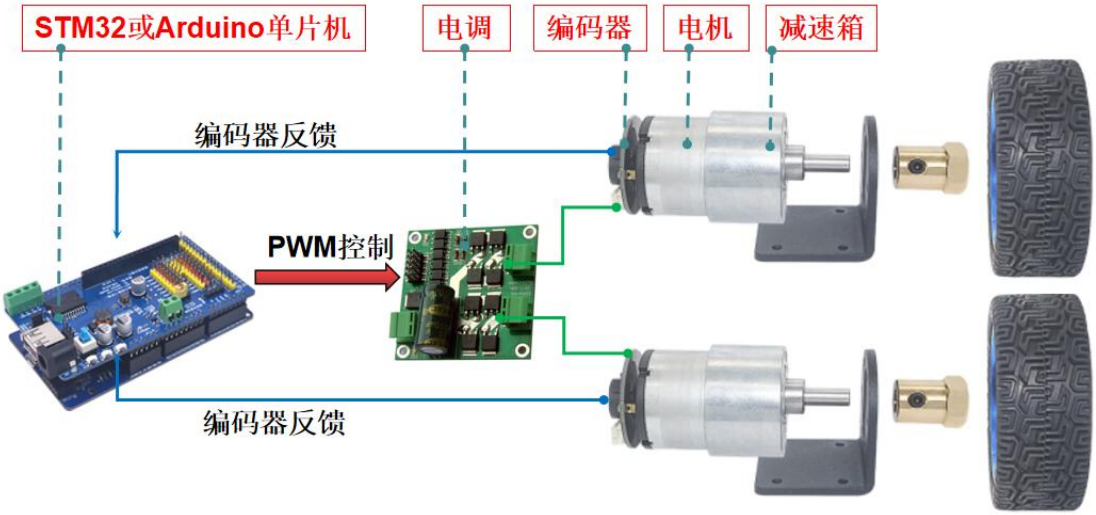

图 2.2.3 小型移动机器人电机选型参考

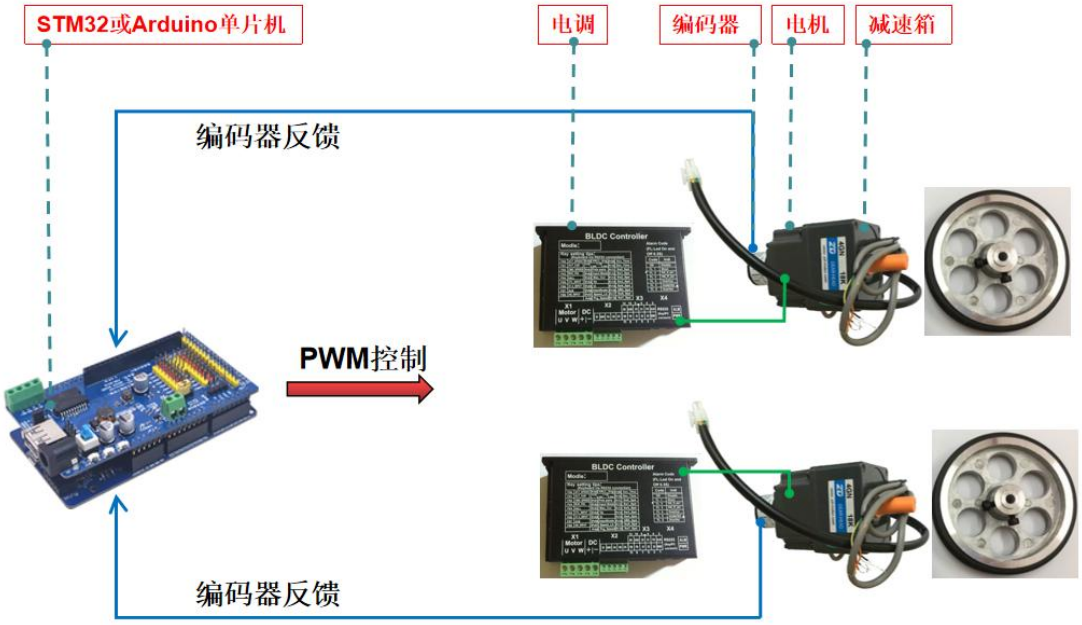

图 2.2.4 载重型移动机器人电机选型参考

2.3.电机驱动板解析

在了解过移动机器人的电机及选型，很自然就会想到接下来用什么来控制电机动起来呢？电机控制需要有 PID 控制，所以电机驱动板主要是用来实现 PID 控制的，同时电机驱动板也负责和上位机通信实时解析上位机发过来的速度指令和反馈电机的速度信息给上位机。电机驱动板一般选用的是 STM32 或者 Arduino 单片机这样的嵌入式 MCU 板子，有朋友可能会问既然都有上位机了（这里的上位机通常是电脑或搭载了操作系统的 ARM 主机）为什么不直接拿上位机来控制电机呢？首先，电机控制 PID 是实时性算法，也就是严格的实时时钟驱动的逻辑，一般的上位机系统不能满足这种高实时性的需求；另外就是出于系统解耦合和模块化设计的考虑，单独用一个价格便宜的 MCU 来实现电机控制。

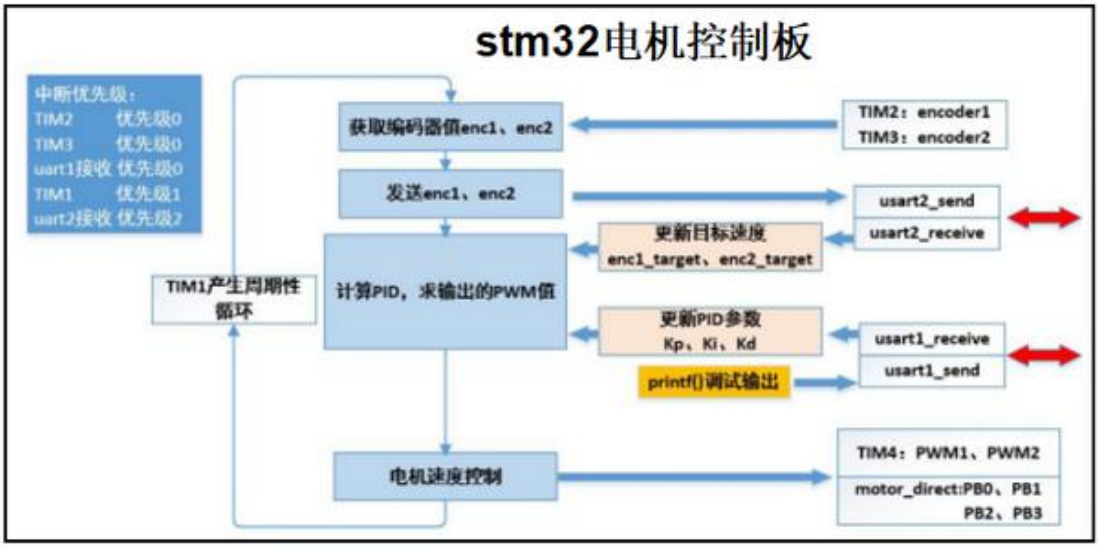

图 2.3.1 一个典型 STM32 电机控制板的程序框图

电机驱动板的另一个重要功能是跟上位机通信，借助上位机上的 ROS 驱动程序实现用

ROS 来操控电机并获取电机编码器反馈的里程计信息。也就是说，电机驱动板+电调+电机组成的一个整体的硬件可以当做是 ROS 机器人的一个传感器来使用，只需要在上位机上启动相应的 ROS 驱动程序，就可以在 ROS 中直接操控电机和获取里程计了。这样的一个效果就是前面提到的系统解耦合与模块化的设计思想，这样的设计对很多不懂硬件的 ROS 开发工程师来说是非常友好的，毕竟不用考虑繁琐的硬件问题了。

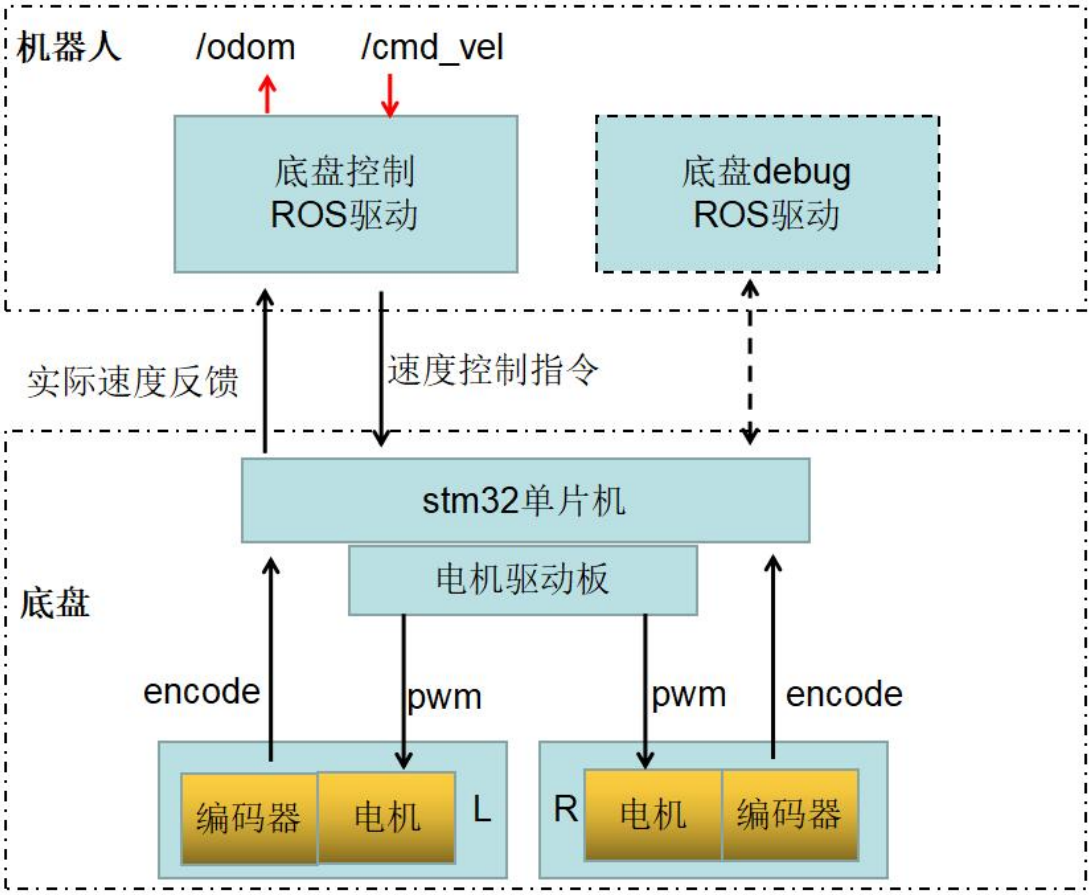

图 2.3.2 在 ROS 中驱动电机控制板

2.4.底盘驱动方式的解析

在了解了电机选型和电机控制板的作用后，底盘驱动方式是移动机器人另一个重要的考虑问题。底盘驱动方式，简单点说就是轮子如何布局和速度合成来让机器人灵活的行走。按照每个轮子的速度输出是否受约束，也就是一个轮子是否可以独立的输出速度而不考虑别的轮子输出速度的约束。约束型的底盘驱动方式有：差分驱动、阿克曼驱动、同步驱动。非完整约束型的底盘驱动方式有：全向驱动、腿足驱动。

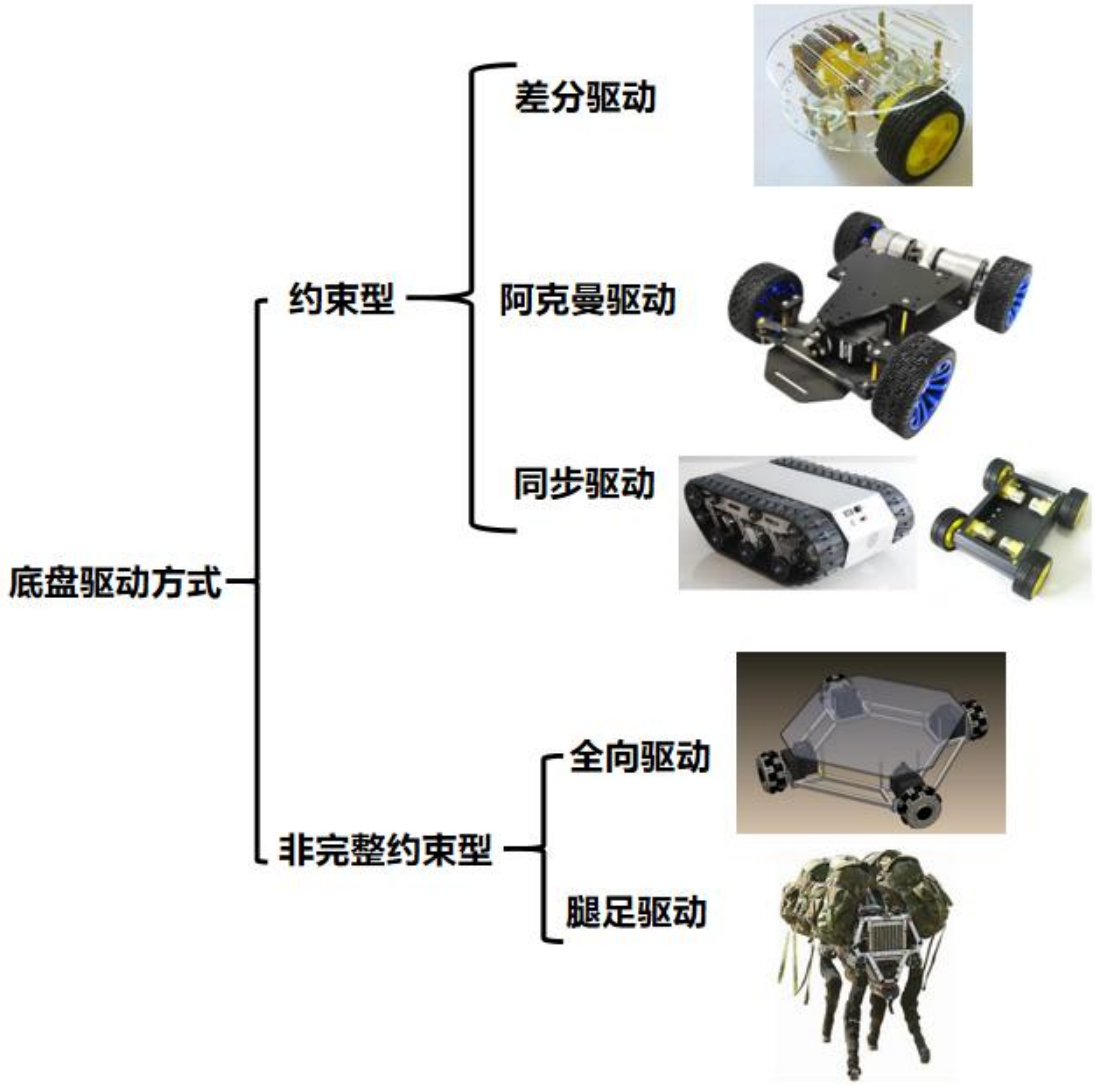

图 2.4.1 底盘驱动方式

差分驱动的底盘，是最常见的形式，像有名的 turtlebot 机器人、pioneer 先锋机器人、家庭扫地机器人都是用的这种驱动方式。差分驱动，其实就是左右两个主动轮输出动力，机器人依靠左右轮的速度差来转弯，这也正是差分驱动名称的由来吧，通过控制左右轮产生不同的差速，来让机器人以不同的线速度和角速度移动。通常，差分驱动底盘还会安装万向从动轮来起到支撑作用。

阿克曼驱动的底盘，在汽车上是常见的形式，绝大部分的汽车底盘就属于阿克曼驱动型。阿克曼车型的两个后轮提供动力输出，两个后轮通过一个差速器来进行速度分配以产生不同转弯半径的差速，前方轮由方向盘控制负责转弯。阿克曼车型的关键就在于后轮的差速器，在汽车上，差速器是一个特殊的齿轮组合结构，能够根据车体实时的转弯半径将发动机的输出速度自动的分配到两个轮子上。而在机器人上，由于驱动轮一般直接连接电动机，所以这里的差速器由电子调速器来替代了。

同步驱动的底盘，最常见的就是履带车了，之所以叫同步驱动，就是同一边的前轮和后轮速度是一模一样的，实在上和差分驱动形式是一样的。有一些没有履带的四轮底盘，也是同步驱动方式。同步驱动的底盘，由于四个轮子，所以载重可以更大也跟平稳。不过相对于差分驱动底盘来说，轮式里程计精度就没那么高了。因为同步驱动的底盘在平坦的路面转弯时，车体会不同程度的颤动。所以同步驱动底盘是不擅长转弯了，转起弯来比较费劲。

全向驱动的底盘，底盘的运动是不受约束的，简单点说就是既可以横着走，又可以直着走，不需要转动车身方向，就可以朝任意方向前进。全向轮为机器人的移动效率起到了很大的提升作用。尤其是像大型货运仓库，AGV 自动物流搬运小车采用全向方式移动能大大节省时间。全向轮是关键部件，全向轮又叫麦克纳姆轮，构造很精巧，不过就是价格昂贵，在非平坦的路面运行起来不那么顺畅了。

腿足机器人，结构复杂，就不在讨论范围了，有兴趣的朋友可以自己上网了解。

2.5.IMU 模块解析

在移动机器人中，IMU 模块也是常用的传感器，IMU 通常被用来和轮式里程计做融合，或者直接参与 SLAM 建图。IMU 是惯性测量单元的简称，用于测量物体的三轴姿态角（roll、pitch、yaw）、三轴加速度（acc_x、acc_y、acc_z）、三轴角速度（w_x、w_y、w_z）等。有六轴 IMU、九轴 IMU、十轴 IMU 等，其实就是 IMU 输出的数据的种类来定义的。

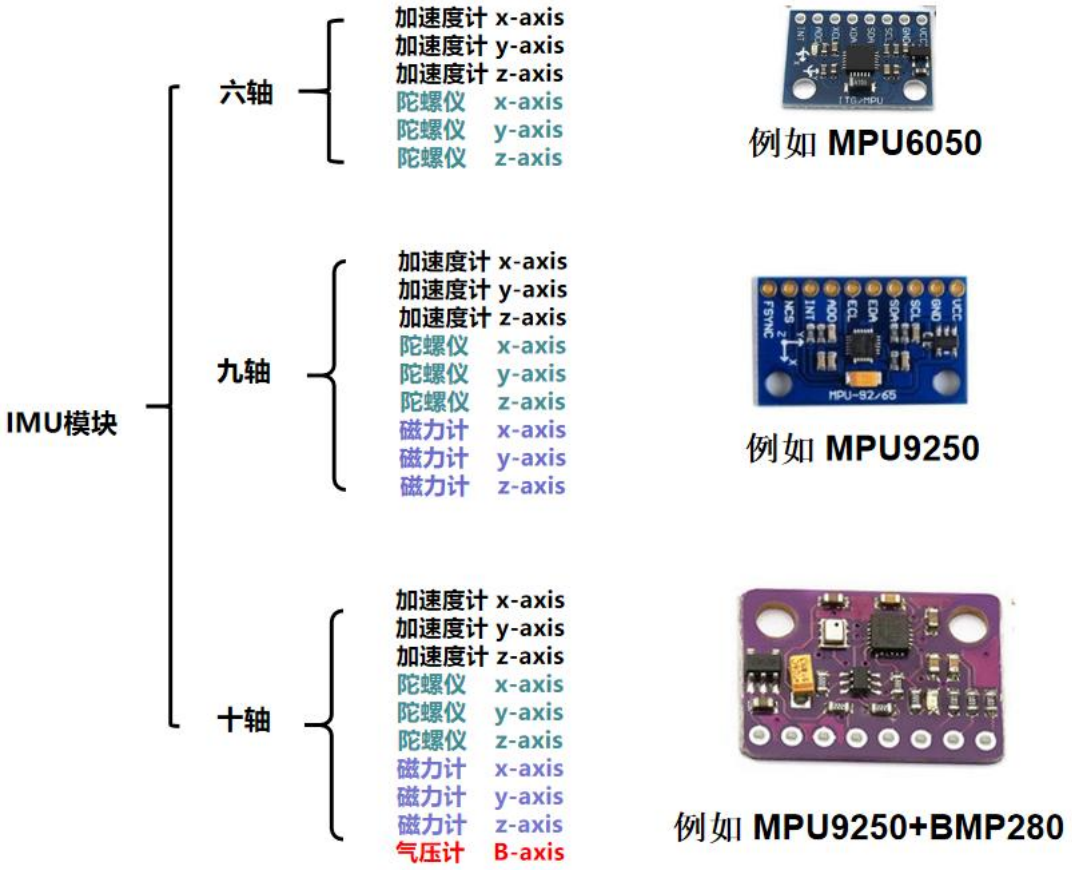

图 2.5.1 六轴、九轴、十轴 IMU 模块

IMU 惯性测量单元在制造过程中，由于物理因素，导致 IMU 惯性测量单元实际的坐标轴与理想的坐标轴之间会有一定的偏差，同时三轴加速度、三轴角速度、三轴磁力计的原始值会与真实值有一个固定的偏差等。这就要求在使用 IMU 模块的原始测量数据时需要对数据进行误差校准，来消除轴偏差、固定零点偏差、尺度偏差等的影响。

acc_calib = Ta*Sa*(acc+Ba+Na)

w_calib = Tw*Sw*(w+Bw+Nw)

mag_calib = Tm*Sm*(mag+Bm+Nm)

mag_calib2 = Tm2a*mag_calib

其中，加速度原始测量值向量 acc 加上零偏补偿 Ba 和白噪声 Na 补偿后，再用尺度矫正矩阵 Sa 和轴偏差矫正矩阵 Ta 变换后，就可以得到校准后的三轴加速度向量 acc_calib；通过类似的方法可以得到校准后的三轴角速度 w_calib 和三轴磁力 mag_calib；最后还要将磁力计坐标系下的磁力数据 mag_calib 变换到加速度坐标系，变换得到的 mag_calib2 将用于加速度/磁力计的数据融合。

图 2.5.2 IMU 误差校准的数学模型

校准过程，其实就是求解误差校准数学模型等式右边的各个校准系数。求解校准系数一般采用最小二次法或矩阵求逆来实现，具体的过程可以参考 IMU 校准方面的论文。

IMU 数据校准完成后，还需要拿 IMU 各个轴的数据进行数据融合来得到三轴姿态角（roll、pitch、yaw）。姿态融合算法有很多种，其中比较流行的有 EKF 姿态融合。

可想而知，要在移动机器人中使用 IMU 模块的数据实现里程计融合或 SLAM 建图，我们需要首先完成更底层的 IMU 误差校准和姿态角融合两个过程。遗憾的是，很多做上层算法开发的小伙伴，对 IMU 模块底层 MCU 级别的程序开发并不擅长，市面上售卖的 IMU 模块大部分都不带误差校准和姿态角融合，有些内置了误差校准和姿态角融合的 IMU 模块价格又太过昂贵。期待不就的将来，能有带误差校准和姿态角融合的 IMU 模块出现，而且价格又能被广大机器人爱好者接受的好产品。

2.6.激光雷达解析

在移动机器人中，获取机器人周围障碍物和环境的轮廓形状是非常重要的。使用激光雷达正是为了实现这个目的。利用扫描得到的障碍物信息，机器人就可以利用 SLAM 建立地图、并进行避障和自主导航。激光雷达之所以流行，得益于它能够精确的测距。主流的激光雷达基于两种原理：一种是三角测距法，另一种是飞行时间（TOF）测距法。其实很好理解，就是利用了最基本的数学与物理知识。

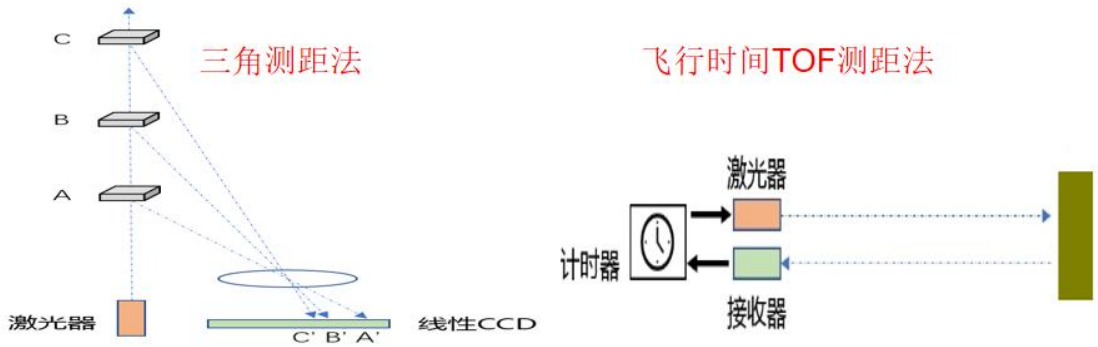

图 2.6.1 三角测距法与飞行时间测距法

激光雷达有单线激光雷达和多线激光雷达，应用场合不一样，价格也差距悬殊。室内移动机器人一般用的是单线激光雷达，价格从几百块到几千块的都有。国内比较早做单线激光雷达的有上海思岚的 rplidar 雷达，后起之秀有深圳镭神智能的 LS 系列雷达、大族激光的杉川系列雷达、深圳玩智科技的 EAI 雷达等等，国外的老牌子雷达就有日本的 HOKUYO 雷达、德国的 SICK 雷达都是很有名的。

室外的无人驾驶汽车一般都配备有多线激光雷达，多线激光雷达一般有 16 线、32 线、64 线、128 线等等。当然线数越多扫描到的信息也就越多，价格自然也就越贵。Velodyne 公司是美国知名的多线激光雷达制造商，其 VL 系列的多线激光雷达也是很有名，价格也是从几万块到几十万不等。国内做多线激光雷达的有速腾、镭神都有在做。

2.7.ROS 主机解析

机器人的处理核心通常是一个运行 ROS 系统的计算机主机，有的复杂一点的机器人还会有好几个计算机主机，比如一个用来搭载 ROS 系统运行各种机器人算法，另一个用来搭载 Android 系统运行用户交互界面。不过这里仅讨论搭载 ROS 系统的主机，有些朋友为了省钱直接用笔记本电脑作为 ROS 主机直接塞进机器人，这种情况除外。搭载在机器人上的 ROS 计算机主机，需要考虑编写安装问题和功耗问题，这里就列举一些常用的 ROS 主机供大家参考吧。

首先是中低端机器人中使用最广泛的树莓派，尤其是树莓派 3B 这个最为流行的型号，当然最近也推出的 3B+升级版本型号。树莓派是极致性价比的板子了，200 块钱的板子，配置了 cortex-A53 的处理器和 1GB 内存，简直就是最便宜的价格最顶配的性能了，并且官方有支持的 ubuntu-mate 系统也是超级给力，所以初学者选用这个板子是非常适合的。

然后中高端机器人可以选择瑞星微 RK3399 这款国产芯片出的板子，萤火虫和友善之臂都有出 RK3399 对应的板子。RK3399 板子价格在 1000 块钱左右，配置了双核 cortex-A72+四核 cortex-A53 的处理器，处理器绝对是给力的，配置的内存 2GB/4GB 可选。搭载裁剪版的 Lubuntu 或 Xubuntu 也是很不错的。

如果是高端机器人可以选择 Nvidia 家的 jetson-tx2 这个板子，这个板子号称是可以搭载在核弹上的板子，可想而知性能是非常牛的。板子售价 4000 块钱左右，有点小贵。配置了四核 cortex-A57 的处理器，256 个 CUDA 核心的英伟达自家的帕斯卡架构 GPU，内存 8GB。板子的亮点是 GPU，非常适合用来做深度学习和计算机视觉方面的应用。

图 2.7.1 ROS 主机

选择好一款适合的 ROS 主机后，就可以在主机上安装 ubuntu 系统了，然后再在 ubuntu 系统上安装 ROS 系统，最后就可以下载各种开源 SLAM 算法开始运行测试了，当然前提是你得有一个能移动的机器人并且各个传感器都是配备好了的。

2.8.摄像头解析

机器人上搭载摄像头，可以用来做视频远程监控、人脸识别和追踪、视觉 SLAM 等等。可以选择的摄像头有单目（mono）、双目（stereo）和深度（RGBD）摄像头。如果是做视觉 SLAM 的朋友，需要对不同摄像头的优缺点有个了解。单目摄像头的优点是结构简单、成本低，缺点是在单张图片里，无法确定一个物体的真实大小。它可能是一个很大但很远的物体，也可能是一个很近很小的物体。通过相机的运动形成视差，可以测量物体相对深度。但是单目 SLAM 估计的轨迹和地图将与真实的轨迹和地图相差一个因子，也就是尺度（scale），单凭图像无法确定这个真实尺度，所以称尺度不确定性。双目摄像头的优点是基线距离越大，能够测量的距离就越远；并且可以运用到室内和室外。缺点是配置与标定较为复杂，深度量程和精度受到双目基线与分辨率限制，视差计算非常消耗计算资源，需要 GPU/FPGA 设备加速。深度摄像头的优点是通过结构光或 ToF(time of fly)的物理方法测量物体深度信息，典型代表 Kinect/xtion pro/RealSense。缺点是测量范围窄，噪声大，视野小，易受日光干扰，无法测量透射材质等问题，主要用在室内，室外很难应用。

图 2.8.1 摄像头

2.9.麦克风解析

对于需要进行语音交互的机器人来说，麦克风是标配。麦克风可以分为阵列形麦克风和普通的麦克风。如果只是简单的录音和识别，选择价格便宜的灵敏度高一点的普通麦克风就可以了。对于语音交互有较高要求，需要做声源定位、回声消除、命令词唤醒这些的话，就必须选择用阵列麦克风了。

3. 商业应用 ROS 机器人

商业应用的机器人形态各异，功能丰富多彩，让人们眼花缭乱，误认为机器人的时代已经到来了。但是大部分市面上能看到的机器人功能相似，智能水平还很有限。一类机器人是主打自主移动导航类型的，比如扫地机器人、送餐机器人、送快递机器人、喷农药机器人等。一类机器人是偏向娱乐性的语音交互机器人，比如桌面型语音助理机器人、智能音响之类的。还有一类机器人是偏内容生产的，就是机器人本体和导航技术语音交互技术都不是自己做的，主要负责生产机器人的内容，比如菜谱机器人、英语早教机器人、会唱歌会跳舞的机器人。下面就结合一些实际的案例，进行详细一点的分析，帮助大家更好的理解 ROS 机器人这个行业的现状。

3.1.SLAM 建图导航的应用案例

（1）智慧农业自动喷药机器人

图 3.1.1 智慧农业自动喷药机器人

这是一个规模化的果园里面，能够自动喷药和配合人工运输采摘下来的果实的机器人。在这样的应用场景，用机器人取代重复繁重的工作是很有意义的。机器人前端安装了一台 3D 激光雷达，能够感知果园道路两旁的果树并进行避障。感兴趣的朋友，可以读一下这篇论文。

图 3.1.2 智慧果园论文

（2）智慧餐厅送餐机器人

图 3.1.3 智慧餐厅送餐机器人

这是一个典型的餐厅场景下自动送餐的机器人，机器人的主要功能就是自主的导航避障，将托盘上的饭菜送达客人的餐桌，配合语音提示和显示屏的交互来满足客人的点餐需求。其实这样的应用场景，机器人采用固定的轨道或磁条导航就已经够用了。毕竟机器人是用来为人类提供服务的，只要能提供稳定高效的服务，不必刻意最求技术有多先进。但是磁条导航也是有缺陷的，就是磁条的铺设比较麻烦，不同的餐厅需要进行不同的铺设，并且磁条时间久了也会损坏需要维修。所以也有很多送餐机器人开始尝试 SLAM 导航了。同样，餐厅服务员的工作也相对繁琐和管理成本高。如果用机器人来替代，不仅可以提高效率，还可以方便用计算机进行统一的管理和调度，非常适合标准化。

（3）楼宇送快递机器人

图 3.1.4 楼宇送快递机器人

说起送快递的机器人，最有需求的莫过于大型电商平台了。以前是消费者在平台上下单，还需要快递员层层转运才能最终送到买家手中。随着物流自动分拣仓库的越来越普及，物流问题最终就集中在最后一公里配送了。如果能用机器人运输快递来解决最后一公里的问题，买家从下单到收到快递完全自动化的了，这个可以说是行业的一种颠覆。公众和企业都在期待呀，作为自建物流配送的京东来说，当然要大力发展快递机器人喽。所以说，快递机器人一旦能成熟的商用，给广大消费者会带来非常大的便利。

（4）大型机房智能巡检机器人

图 3.1.5 大型机房智能巡检机器人

一个大型的机房，有成百上千台服务主机。要实时监测每台主机的各项参数，比如主机的温度、网络连接状态、电源指示灯等是否正常。在这样一种高噪音高辐射的恶劣环境里，用人来完成日常巡检工作异常困难。所以，利用自主移动机器人配合机器视觉等机会就能解决不少问题。类似的场景还有图书馆、变电站、甚至核电站。

3.2.商业 ROS 机器人相关公司

（1）日本软银集团 pepper

软银的 pepper 是一款很有名的人形机器人了，在展会上看到过，手臂挥舞配合说话的语音腔调，还蛮有人情味的。这款机器人主要是用在服务业和零售业的吧，比如说当迎宾和解说之类的。

（2）北京进化者机器人公司的小胖

小胖机器人的用户体验和功能算是做的非常好的了，至少来说算是真正能为人类做点事情的。比如投影仪功能、空气净化功能、配孩子写作业。机器人这个行业要能做到这些功能

还要控制好价格真的太不容易了。而且小胖机器人定位 2~14 岁孩子这样一个用户群体，定位很好，至少现阶段还很初级的人工智能技术也只能在小孩子中发挥作用。虽然说人们想象中的完全智能的机器人还很遥远，但在可控范围内提供特定功能的机器人还是由不少的用处的。小胖机器人也是为数不多能量产的家庭服务机器人了吧。

（3）上海思岚科技的 ZEUS

思岚是做激光雷达的，有名的 rplidar 大家都很熟悉，依托自家的激光雷达，同时开发了配套的 SLAM 导航算法板，陆续又推出了 APOLLO 和 ZEUS 系列的移动底盘解决方案。想往 SLAM 导航方案公司发展吧，导航方案的具体效果嘛这个还不好说，毕竟整个行业目前还没什么大的突破。

（4）上海高仙机器人公司的室外机器人

高仙机器人主打的是室外环卫清洁机器人、安防监控机器人、楼宇物流机器人。专注于室外的 SLAM 导航技术，SLAM 导航方案在行业内做的相对也是不错的。

（5）云迹科技的大润机器人

云迹专注的是酒店服务员机器人这样一个特定场景，帮助入住酒店的顾客进行导引和提供递送饮料毛巾之类的服务。机器人可以帮助顾客智能开锁、呼叫前台、酒店资讯查询。亮点之一是可以自动上下电梯，这个在酒店应用环境真的很实用。在酒店这样一个相对稳定简单的场景下，SLAM 导航工作还是比较稳定的，这也是云迹着眼的比较好的一个点。

（6）交通银行大堂机器人娇娇

娇娇在网上火了一把，吸引了不少流量。娇娇的最大特点是幽默风趣的聊天技巧和卖萌的表情。有很多人怀疑机器人后面是不是有人工控制，不然以当前的人工智能水平怎么可能达到这样逼真的效果。技术问题我们暂且不用去纠结，毕竟娇娇这样一个走红的机器人确实能说明机器人在一些方面确实能起到作用。

3.3.商业 ROS 机器人两大发展思路

机器人几大核心技术还没有明显突破，众多资本进入这个行业，导致了各种机器人是同质化严重，这个本来未成熟的蓝海变成了红海。那么各大机器人公司都是怎样的发展思路呢？这里就做一些分析吧，下面主要介绍两种发展思路：核心部件和系统集成。

（1）核心部件

一开始机器人这个行业前途明朗的情况下，很多创业公司选择机器人上的摸个重要部件作为公司的主打产品，一方面是降低风险，另一方面也算是发展自己的核心技术优势吧。

激光雷达就属于其中一种核心部件，比如思岚、镭神、北醒都是从激光雷达传感器开始做起来的，等到激光雷达有市场了，就开始依托自家的激光雷达来开发配套的 SLAM 导航方案，慢慢的就开始给客户提供个性化的定制机器人方案了。

移动底盘也属于其中一种核心部件，比如深圳玩智科技、高仙机器人都是从移动底盘开始做起的。做底盘以后可以朝无人驾驶方向发展，另外就是这个环节是机器人集成设计中能分割出来的比较大的一块，并且是有技术含量的，最重要的是能够下的了手开展工作的。像有些机器人上的核心技术，一般的创业公司根本没有能力自己做。自己的底盘做好了，还可以推出配套的 SLAM 导航解决方案，甚至还可以和下游商合作推出实际的产品。

所以说，从核心部件开始做机器人，既能够降低风险和出成果，又能够积累自己的核心技术将来向系统集成方向发展。但是，不是任何公司都能做核心部件的，毕竟核心部件涉及到太专业的领域，没有专业背景强大的团队，根本沾不上边。

（2）系统集成

由于资本的力量，很多没有研发能力的公司也向做机器人怎么办？就买别人的核心部件，

然后自己拼起来在上面做一些自己的应用了。这就会系统集成型的公司，系统集成型的公司要做的好的话，严格上来说需要更大规模的团队和实力。因为系统集成考虑的问题更多了，要做好各个方面的用户体验和稳定性。但是目前的大多数系统集成都是简单的组装，有些集成甚至干脆就找外包了，自己只负责业务销售了。可想而知，没有亮点的同质化机器人最终是做不长久的。于是很多系统集成的机器人公司在发展无果的情况下，又纷纷转向核心部件的研发。当然也有一些专注特定领域产品定位的机器人公司做了一些出色的产品了，比如小胖、云迹。稍有实力的公司，咬咬牙，就算是养团队也要自己做底盘、雷达、语音交互这些核心部件。因为如果产品真的要量产，这些技术必须掌握在自己手上。

4. 科研学习 ROS 机器人

4.1.科研学习 ROS 机器人与商业应用 ROS 机器人的区别

商业 ROS 机器人都是面向具体的产品落地，或者直接面向直接的客户开发设计的。比如说外观，商用机器人会考虑美观实用性，传感器的摆放也会根据具体应用场景有针对性的设计。机器人上还会考虑自动充电、紧急情况下应急开关、碰撞保护等实际问题。并且考虑到产品的稳定性和成本，往往硬件接口设计的会更加紧凑和耦合，目的就是节省成本和体积。软件设计方面，也会考虑跟多运行效率的东西，往往会对 ROS 系统进行必要的裁剪，并且设计一些自己专用的通信协议。商业 ROS 机器人的这些优点，对于学习和科研用途的朋友来说刚好是缺点。因为外观设计使得机器人外形结构相对固定了，不利于 ROS 学习开发者安装自己的传感器进行调试和实验；紧凑和耦合的硬件接口设计，不利于 ROS 机器人学习者对机器人整体工作原理进行理解和后续二次开发；软件设计上的裁剪和特殊协议设计，也不利于代码标准化和 ROS 官方 DEMO 的测试。

科研学习 ROS 机器人正是为了学习开发者而设计的，所以机器人外观相对简单并且具有极强的可拼接性，开发者可以根据自己的需要通过简单的改造就能搭建自己需要的外观的机器人。机器人上的硬件设计也更加的模块化，电机控制、IMU、激光雷达、摄像头、麦克风基本都是独立标准的外设，并且大多数都提供配套的 ROS 驱动程序，让学习者只有删减传感器和修改参数也耿便利。软件设计也大多使用各种流行的开源代码，资料相对丰富，基本上接口也都是开放的。

所以做机器人 SLAM 导航和各种相关应用开发的朋友来说，选购一款适合自己的 ROS 机器人学习平台就非常重要了。当然如果自己动手能力强，时间又比较充足，又具备硬件电路设计、结构设计、单片机开发、电机控制调试、还懂上位机 ROS 驱动程序开发，可以自己设计一套机器人底盘出来。但是毕竟这个级别的大神数量是很有限的，所以选择一个 ROS 机器人学习平台就很必要了。

4.2.淘宝上科研学习 ROS 机器人相关产品

（1）Pioneer 先锋机器人

大名鼎鼎的 Pioneer 先锋机器人做 ROS 机器人的都应该知道，这款机器人在市场上卖了都快 20 年了，Pioneer3-DX 这个型号在全世界各大科研机构很受欢迎。Pioneer 3-DX 是一款耐用的、差分驱动的机器人，它主要应用在教学研究上。Pioneer 3-DX 在使用上具有较高可靠性和耐用性，在开发应用上具有多功能性，这使它在教学研究领域成为最受欢迎的差分驱

动移动机器人。MobileRobots 公司专业的设计和加工能力，使 Pioneer 3-DX 能够在教室或实验室环境中正常使用几年的时间。Pioneer 3-DX 装配有 500 线编码器的电机、19cm 的轮胎、铝制外壳、8 个前置防碰撞声纳，根据需要还可为用户安装 8 个后置防碰撞声纳。Pioneer 3-DX 可以安装 3 块热拔插电池，除了以上硬件我们还为用户提供整套的开发软件。现在，只需再为机器人安装上车载工控机或笔记本电脑就可以让它走起来。

图 4.2.1 Pioneer 3-DX

先锋机器人是很好，缺点就是太贵了，而且核心的模块和传感器都没有太多的资料公开。还有就是外国人用的多，在国内还需要专门引进。不过优点就是很耐用，勇哥三到五年没什么问题。

（2）Turtlebot 海龟机器人

另一款机器人 Turtlebot 大家应该就更加熟悉了，俗称小海龟机器人，是 ROS 官方推出的一款移动机器人，ROS 系统中很多默认的程序例子都是以小海龟机器人为原型的，所以说小海龟在 ROS 机器人中的重要地位了。

图 4.2.2 Turtlebot3

小海龟机器人外观合计还是很精致的，毕竟是 ROS 官方强烈推荐的。小海龟机器人在 wiki 和 github 的软件资料和教程也是相当丰富的。但是 5000RMB 的价格对于国内的众多 ROS 机器人爱好者来说还是很难接受，虽然相比 Pioneer 机器人已经便宜很多了。另一个就是，教程都是英文的，中国读者还是有一定的不方便的。

（3）EAI 移动底盘

这是中国深圳的一家专业做 ROS 移动底盘的公司，底盘还是相对性能稳定的，尤其是 50kg 大载重的性能很有亮点。EAI 的底盘大部分是卖给做二次开发的用户的吧，学习 ROS 机器人的开发者比较少。是由于 EAI 的底盘基本上就是一个移动底盘，不带上层的 SLAM 建图导航和配套的学习资料。将近 1 万块钱的售价，如果作为 ROS 机器人初学者买回去也是不划算的。但是需要大载重的底盘，这个价格还是可以接受的。这里一直在提各个 ROS 机器人价格贵，不是说价格便宜的机器人就一定好。如果项目资金充足，性能上确实有刚需，肯定是推荐买贵的能满足需求的 ROS 机器人。

（4）miiboo 语音交互移动机器人

这也是深圳的一家专业做 ROS 机器人的公司，跟 EAI 的底盘恰恰不同，miiboo 机器人主

打的是 ROS 教育入门。miiboo 机器人的硬件设计的模块化做的很好，电机控制、IMU、激光雷达、摄像头、麦克风、音响都是独立设计，并配有很详细的硬件传感器文档及 ROS 驱动程序。

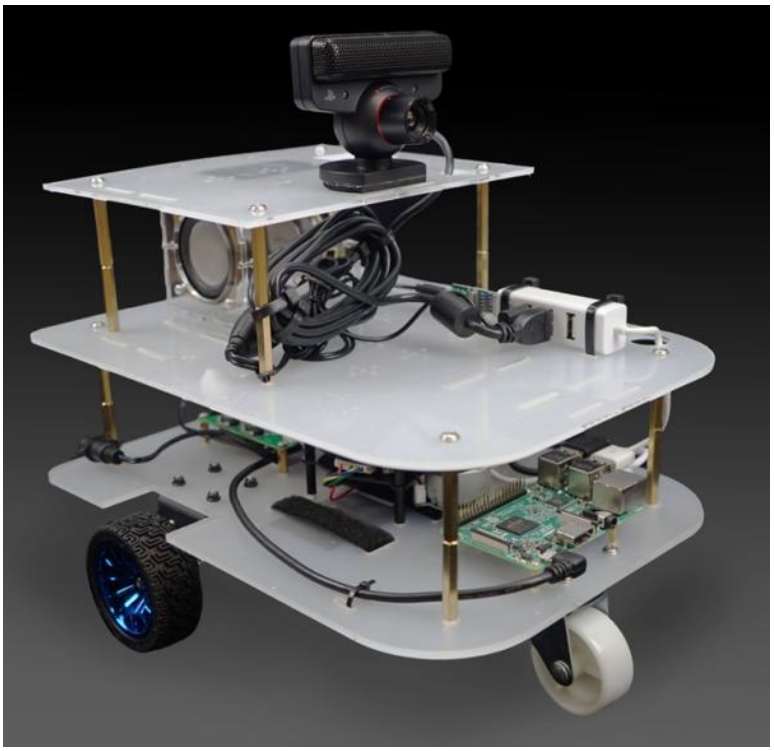

图 4.2.3 miiboo 机器人

miiboo 机器人的亮点主要由两个，一个亮点是 SLAM 导航算法做的效果非常好，用的是 google 最新版本的 cartographer 建图算法，配合轮式里程计、IMU、激光雷达三者融合建图，并且 SLAM 导航功能基础上还支持语音识别、语音合成、语音交互功能，SLAM 导航+语音交互的结合已经很接近一个实际的机器人了；另外一个亮点就是 miiboo 机器人的 ROS 开发文档做的相当棒。算是良心做 ROS 机器人教育的一家了吧。开发资料做的很系统，既有最基础的 Linux、ROS 方便的编程入门，也有底盘电机控制、单片机硬件软件、PID 控制、里程计标定方面的核心技术细节讲解，还有高级的 cartographer 算法核心分析和参数调参指导，还有语音交互和机器人手机 APP 开发方面的教程。这么系统的教程，真的比有些宣传自己的 ROS 机器人提供大量开发资料的要强很多。而且真的是良心团队，这些系统的开发资料都全部公开在了博客上（https://www.cnblogs.com/hiram-zhang），就算不买他家的 miiboo 机器人，也能够免费学习全套的资料，足以见得 miiboo 机器人团队的自信。并且博客中网友的提问，他们也是很认真的倾听和解答，并很快在 miiboo 机器人做出改进，这个必须给赞。

图 4.2.4 miiboo 机器人配套的教程

这个机器人是主打 ROS 开发入门和 ROS 教育的，所以如果是大载中需求的应用车体估计就够呛了。

（5）淘宝上眼花缭乱的 ROS 机器人

然后，就是淘宝上眼花缭乱的 ROS 机器人了，形态各异。比如说有履带底盘的，看起来还是很酷的，不过要是真的做起 SLAM 导航和建图算法来，估计精度应该还是会有比较大的出入的，站在专业的角度讲，其实履带底盘的里程计模型并不好，也不被主流机器人所接受，如果真的要在 SLAM 算法上有所突破和成果，还是推荐选择主流的两轮差分驱动的底盘机器人，里程计精度应该是最高的。还有一些做的也很精美的机器人，还配备有深度相机和高端的主机，这个因人而异，但是作为推荐还是不推荐买这种少见的不流行的机器人。

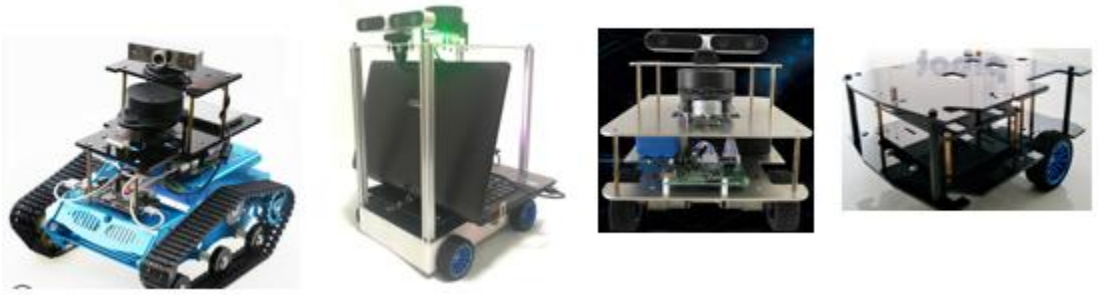

图 4.2.5 眼花缭乱的 ROS 机器人

4.3.选购 ROS 机器人入门指南

（1）提供系统的开发和使用教程的重要性

相信大部分购买 ROS 机器人学习的朋友，都是想通过机器人学习整个 ROS 的软硬件构造和工作流程，然后有能力设计自己的机器人，或者按照自己的算法对机器人做出必要的改造与二次开发。所以一款 ROS 机器人是否公开硬件设计及原理、软硬件的工作流程、已经 ROS 软件驱动程序的代码及参数作用，这些关键的资料就至关重要了。学习 ROS 机器人开发的朋友们基础水平肯可能各不相同，有些爱好者可能是从 Android 或者 java 转行过来搞 ROS 机器人的，所以对 ROS 系统基础并不多；有些爱好者可能最开始是搞硬件单片机开发的，对 Linux 这些系统级的上层开发基础比较差；还有些爱好者是计算机专业出身，以前只搞过算法和框架开发，底层的东西不熟。这里不是说要求所有的 ROS 机器人开发者都对机器人上的所有部分在行，但是要在机器人上做出好东西，就要有系统全盘的了解。所以一款 ROS 机器人能否提供从底层到上层的系统学习教程非常关键。其实，Turtlebot 小海龟机器人和 miiboo 机器人在这方面做得都算不错，如果资金充足可以买 Turtlebot 小海龟机器人，如果资金有限制可以买相对便宜一些的 miiboo 机器人。

（2）外形结构可扩展的重要性

其实这个大部分的机器人做的都是可以的，基本上是三层板垒起来的结构都是符合扩展要求的。

（3）硬件模块化的重要性

这个必须提一下，之前用过一款叫 irobot 的底盘，就是扫地机器人那种。硬件接口全部集成隐藏在塑料壳子里面，也不知道里程计、IMU 是怎么安装和工作的，只有一个简单的文档告诉你怎么驱动它。每次没电了，这个底盘就在那叫，还乱撞，感觉用户体验很不好，而且模块都封装在塑料壳子里面，对开发者很不友好。

（4）别太迷信软件开源

见过很多 ROS 机器人上宣传说提供各种高级的图像识别算法、自动跟踪算法、有些还开源底层的单片机程序、开源导航算法等等。听上去有很多功能开源，但是大部分是本来就开源的。有一些像电机控制的单片机程序或 IMU 模块单片机程序开源，但是本身底层的这些固件程序要做到效果好的很难，开源往往意味着效果并不怎么好，甚至代码格式写的是很糟糕的。所以其实不要太迷信软件开源这个噱头，能提供软件源码及在机器人上的详细教程的才是靠谱的，所以一定要看这些宣称提供软件开源的 ROS 机器人是否提供了足够有含金量的免费教程供大家检验。

（5）技术支持的重要性

作为开发学习用途的 ROS 机器人，出现各种技术问题和 bug 是再正常不过的事情了。

一般技术开发，都有很多论坛和社区来提供问题的交流和解答。但是 ROS 机器人本来就是最近才兴起的，而且国内的 ROS 机器人还没有发展出规模，所以其实还不存在特别专业靠谱的技术论坛。所以，购买了 ROS 机器人，会否能得到专业的技术支持至关重要。由于提供专业的技术支持人力成本非常高，所以一般的公司根本负担不起免费的技术支持服务。所以，一家 ROS 机器人是否提供免费专业的技术支持也会相当的重要。

后记

为了防止后续大家找不到本篇文章，我同步制作了一份文章的 pdf 和本专栏涉及的例程代码放在 github 和 gitee 方便大家下载，如果下面给出的 github 下载链接打不开，可以尝试 gitee 下载链接：

- github 下载链接：
  https://github.com/xiihoo/DIY_A_SLAM_Navigation_Robot

- gitee 下载链接：
  https://gitee.com/xiihoo-robot/DIY_A_SLAM_Navigation_Robot

参考文献

张虎,机器人 SLAM 导航核心技术与实战[M].机械工业出版社,2022.