**基础语言学习——**

**C++——（SLAM与多传感器融合）**

**需要学习的部分：**

* **基础语法：变量、数据类型、控制结构、函数、指针、引用等。**
* **面向对象编程（OOP）：类、继承、多态、封装等。**
* **标准模板库（STL）：容器（如 vector, map, set）、算法（如 sort, find）等。**
* **内存管理：动态内存分配、智能指针（如 unique\_ptr, shared\_ptr）。**
* **多线程编程：std::thread, std::mutex, std::condition\_variable 等**

**相关的AMR感知算法库：**

 **OpenCV：用于图像处理和计算机视觉。**

** PCL (Point Cloud Library)：用于点云处理。**

** ROS (Robot Operating System)：广泛用于机器人开发，C++ 是其主要支持语言之一。**

** g2o、Ceres Solver：用于优化问题，常见于 SLAM 中**

**STL标准库学习导向——**

**1，ROS导航栈：基于多线程的roscpp框架，使用std::vector存储点云数据，std::thread分离定位与规划任务。**

**2，运动规划库（OMPL）：利用STL容器管理状态空间，多线程加速采样**

**多线程编程导向——**

**1，异步数据流：例如，一个线程处理SLAM（同步定位与建图），另一线程执行运动控制。**

**2，计算密集型任务并行化：将路径搜索算法（如A\*、RRT\*）拆分为多线程子任务，加速解算。**

**Python——（构造深度学习模型）**

**科学计算库导向——高效处理这些****多模态传感器数据**

* **NumPy：多维数组操作处理激光雷达点云（3D坐标）、图像（RGB矩阵）、IMU时序数据。**
* **OpenCV：图像处理：去畸变、特征提取（如 ORB、SIFT）、立体视觉匹配。 视频流处理：实时摄像头数据的读取、裁剪和增强。**
* **Pandas：处理传感器标定参数、轨迹日志、环境特征表格。**

**深度学习框架导向：**

* **TensorFlow ，PyTorch：用于构建和训练深度学习模型。**
* **模型搭建**：设计用于 SLAM（如深度学习特征匹配）、路径规划（如强化学习策略网络）或环境感知（如语义分割）的神经网络。
* **自动微分**：通过反向传播优化模型参数，例如训练一个端到端的视觉导航模型

**可视化工具导向：**

** Matplotlib/Seaborn：绘制损失曲线、特征分布图、路径规划结果。**

** TensorBoard：监控模型训练过程**

**优先掌握：NumPy（数据操作）、OpenCV（图像处理）、PyTorch/TensorFlow（深度学习）。**

1. **实战项目：**
   * **用 Python 实现一个简单的视觉 SLAM 前端（特征提取+匹配）。**
   * **训练一个 DRL 模型在仿真环境中完成避障任务。**

**感知算法库学习——**

1. **PCL——激光雷达点云的滤波、分割、配准、特征提取**
2. **OpenCV——图像处理、特征提取、目标检测**
3. **LIO-SAM——激光惯性里程计**
4. **EKF滤波 粒子滤波——处理融合传感器数据，减少噪声，提高估计运动状态的精度**