

# ROS机械臂开发：从入门到实战

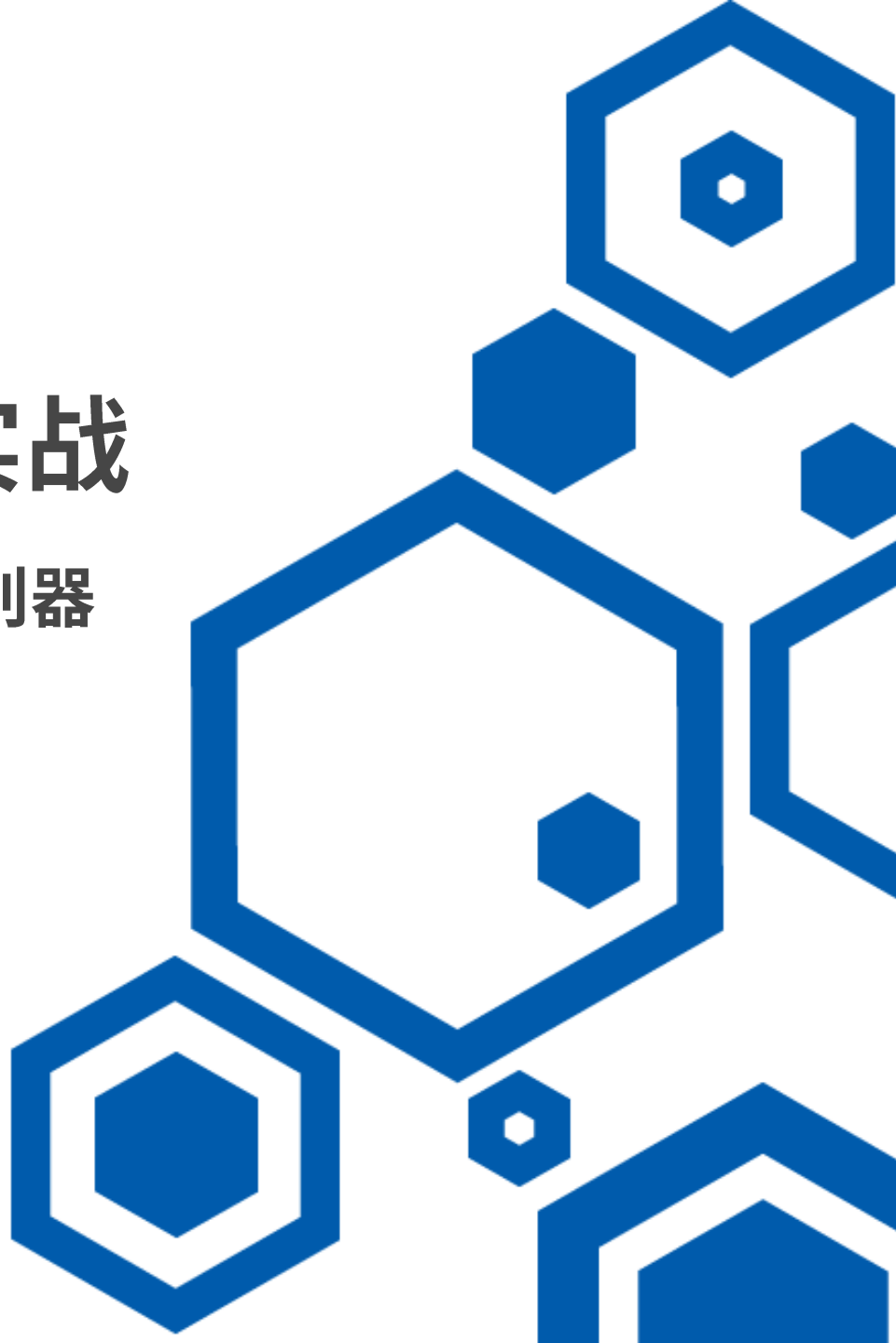
## —— 第12讲：ROS——机器人开发的神兵利器



主讲人 胡春旭



机器人博客“古月居”博主  
《ROS机器人开发实践》作者  
武汉精锋微控科技有限公司 联合创始人  
华中科技大学 自动化学院 硕士



 1. 课程总结

 2. 进阶攻略

 3. 资源整理



# 1. 课程总结



## 1. 课程总结

# ROS机械臂开发：从入门到实战

### 基础原理篇

#### 1. ROS的过去、现在和未来

- 1.1 ROS发展与现状
- 1.2 课程介绍

#### 2. 风靡机器人圈的ROS到底是什么

- 2.1 通信机制
- 2.2 开发工具
- 2.3 应用功能
- 2.4 社区生态

#### 3. 如何从零创建一个机器人模型

- 3.1 URDF建模
- 3.2 Solidworks导出模型

#### 4. ROS机械臂开发中的主角MoveIt!

- 4.1 MoveIt!简介
- 4.2 MoveIt!可视化配置助手

### 功能实践篇

#### 5. 搭建仿真环境一样玩转ROS机械臂

- 5.1 ROS中的控制器插件
- 5.2 构建MoveIt!+Gazebo仿真

#### 6. MoveIt!编程驾驭机械臂运动控制

- 6.1 关节空间运动
- 6.2 笛卡尔空间运动
- 6.3 自主避障运动

#### 7. MoveIt!中不得不说的“潜规则”

- 7.1 圆弧运动规划
- 7.2 轨迹重定义
- 7.3 多轨迹连续运动
- 7.4 更换运动学插件

#### 8. ROS机器视觉应用中的关键点

- 8.1 ROS图像接口
- 8.2 摄像头内参标定
- 8.3 物体识别案例分析

### 综合应用篇

#### 9. “手眼”结合完成物体抓取应用

- 9.1 手眼标定
- 9.2 机械臂抓取

#### 10. 针对工业应用的ROS-I又是什么

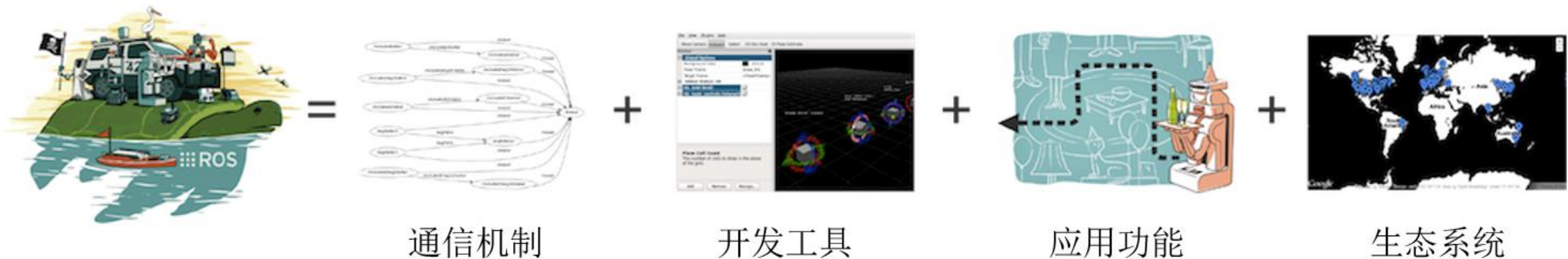
- 10.1 ROS-I框架介绍
- 10.2 ROS-I应用原理
- 10.3 ROS-I代码浅析

#### 11. 基于ROS设计一款机械臂控制系统

- 11.1 ROS控制系统设计方法
- 11.2 PROBOT Anno控制系统案例分析

#### 12. ROS — 机器人开发的神兵利器

- 12.1 课程总结
- 12.2 进阶攻略
- 12.3 资源整理



## ROS 是什么



### 通信机制

松耦合分布式通信：节点、管理器、话题、服务 …



### 开发工具

命令行、Launch、TF、Qt工具箱、Rviz、Gazebo …



### 应用功能

Navigation、SLAM、MoveIt! …



### 生态系统

发行版、软件源、wiki、ROS Answers …

## URDF 建模

- 机器人组成：执行机构，驱动系统，传感系统，控制系统
- URDF文件中的标签：<link>、<joint>、<robot>
- 创建机器人URDF模型：设计外观（link），拼装集成（joint）

## 机械臂 建模

- 六轴机器人：7 links + 6 joints
- 可视化显示：模型路径 + joint\_state + tf + rviz

## 三维模型 导出

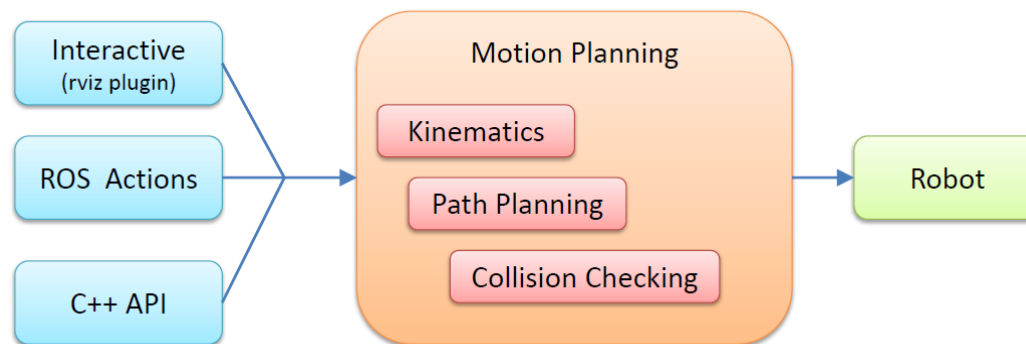
- 完成模型设计，安装sw2urdf插件
- 根据说明，完成link和joint的配置
- 检查并确认配置，生成模型功能包
- 修改文件bugs，测试模型



# 1. 课程总结 —— 4. ROS机械臂开发中的主角MoveIt!

## MoveIt! 简介

- 一个易于使用的集成化开发平台



## MoveIt! 可视化配置

- 启动Setup Assistant
- 加载模型
- 配置自碰撞检测
- 配置规划组
- 预定义机器人位姿
- 输入作者信息
- 自动生成配置包
- 运行demo.launch进行测试



# 1. 课程总结 —— 5.搭建仿真环境一样玩转ROS机械臂

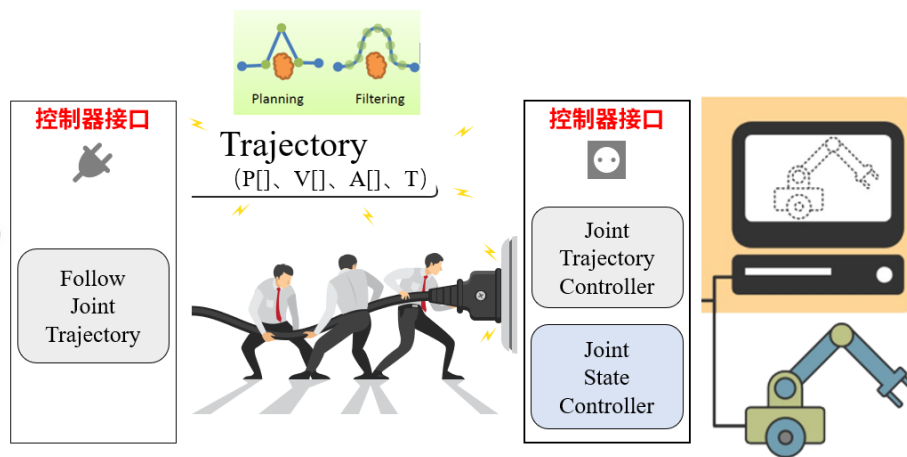
## ROS中的 控制器插件

- ROS为开发者提供的机器人控制中间件
- 包含一系列控制器接口、传动装置接口、硬件接口、控制器工具箱等等
- 可以帮助机器人应用功能包快速落地，提高开发效率

## 完善机器人 模型

- 为link添加惯性参数和碰撞属性
- 为joint添加传动装置
- 添加gazebo控制器插件

## 构建MoveIt! +Gazebo仿真







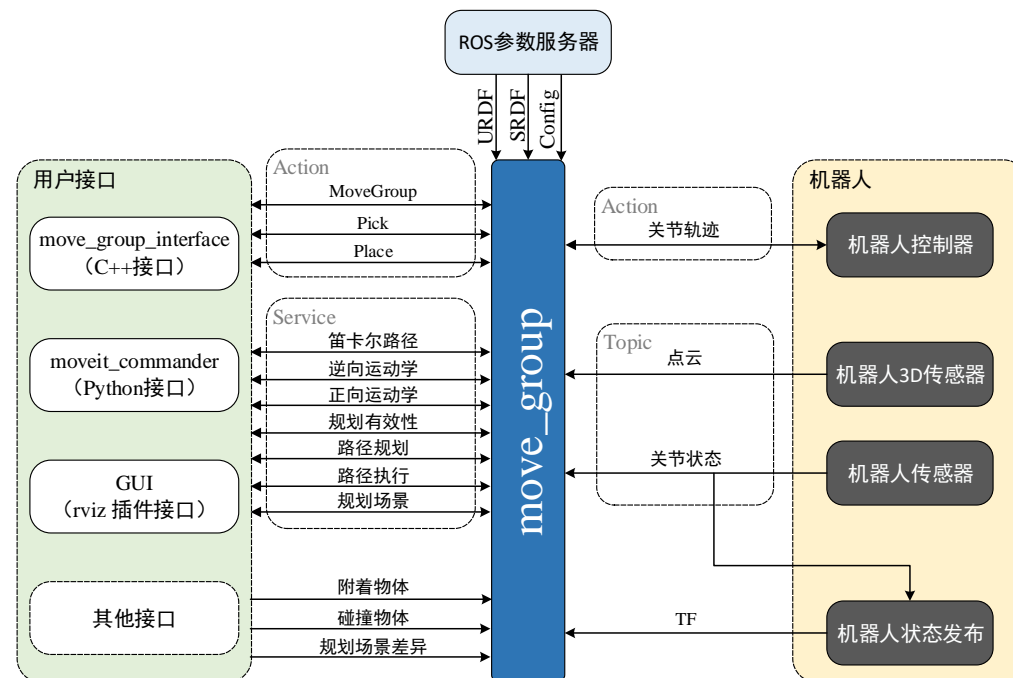
# 1. 课程总结 —— 6. MoveIt!编程驾驭机械臂运动控制

## MoveIt!的编程接口

- C++、Python
- 编程流程：规划组→目标位姿→运动规划→运动执行

## MoveIt!基础编程

- 关节空间运动
  - set\_joint\_value\_target
  - set\_pose\_target
- 笛卡尔空间运动
  - compute\_cartesian\_path
- 自主避障规划
  - rviz添加障碍物
  - add\_box
  - attach\_box





# 1. 课程总结 —— 7. MoveIt!中不得不说的“潜规则”

## 圆弧轨迹规划

- 计算圆弧轨迹 → 规划笛卡尔路径 → 执行轨迹运动

## 轨迹重定义

- 规划轨迹 → 轨迹重定义 → 执行轨迹运动

## 多轨迹连续运动

- 计算多条轨迹 → 轨迹拼接 → 重新规划速度、加速度 → 执行轨迹运动

## 更换运动学插件

- KDL：易使用，但失败率高、效率低
- TRAC-IK：成功率高，但求解不稳定
- IKFAST：成功率高、求解稳定、速度快，但存在多解选择问题



# 1. 课程总结 —— 8. ROS机器视觉应用中的关键点

## ROS图像接口

- 二维图像：sensor\_msgs/Image、sensor\_msgs/CompressedImage
- 三维图像：sensor\_msgs/PointCloud2

## 摄像头 内参标定

- 避免摄像头内部光学器件造成的数据源畸变误差
- ros-melodic-camera-calibration

## ROS+OpenCV 物体识别

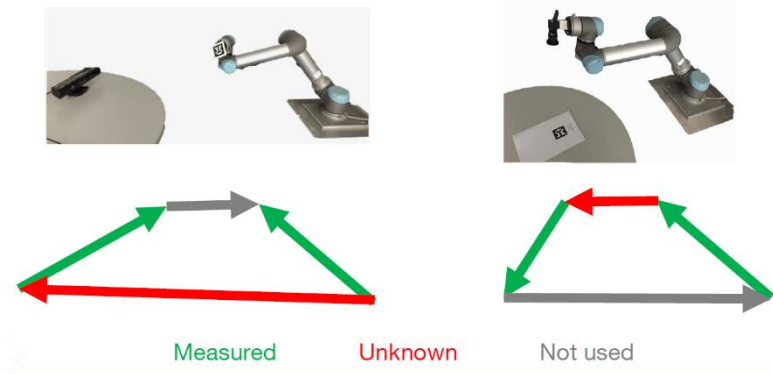
- CvBridge：转换ROS与OpenCV之间的图像数据
- 物体识别流程：
  - ROS驱动摄像头，发布图像消息
  - 将ROS图像消息转换成OpenCV图像数据
  - OpenCV图像处理
  - OpenCV图像转换成ROS消息

## 视觉抓取中的关键技术

- 手眼标定：相机内参、外参的标定
- 物体识别与定位：物体在什么、物体在哪里
- 抓取姿态分析：如何完成抓取动作
- 运动规划：轨迹规划、运动学、动力学、碰撞检测

## 手眼标定

- 相机位置：
  - eye to hand 眼在外
  - eye in hand 眼在手



## 机械臂视觉抓取案例

- 机器人坐标系、图像坐标系、相机坐标系、工件坐标系之间的关系
- 视觉识别与运动控制的集成

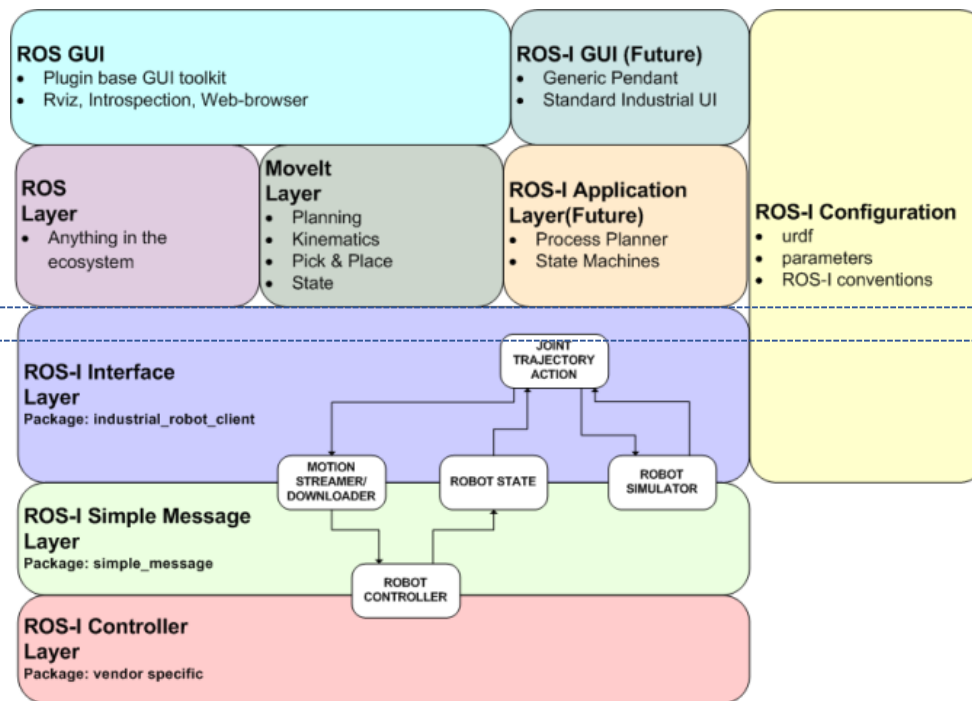


# 1. 课程总结 —— 10.针对工业应用的ROS-I又是什么

## ROS-I框架介绍

- 将ROS强大的功能应用到工业生产的过程中；
- 为工业机器人的研究与应用提供快捷有效的开发途径；
- 为工业机器人创建一个强大的社区支持；
- 为工业机器人提供一站式的工业级ROS应用开发支持。

## ROS-I应用原理

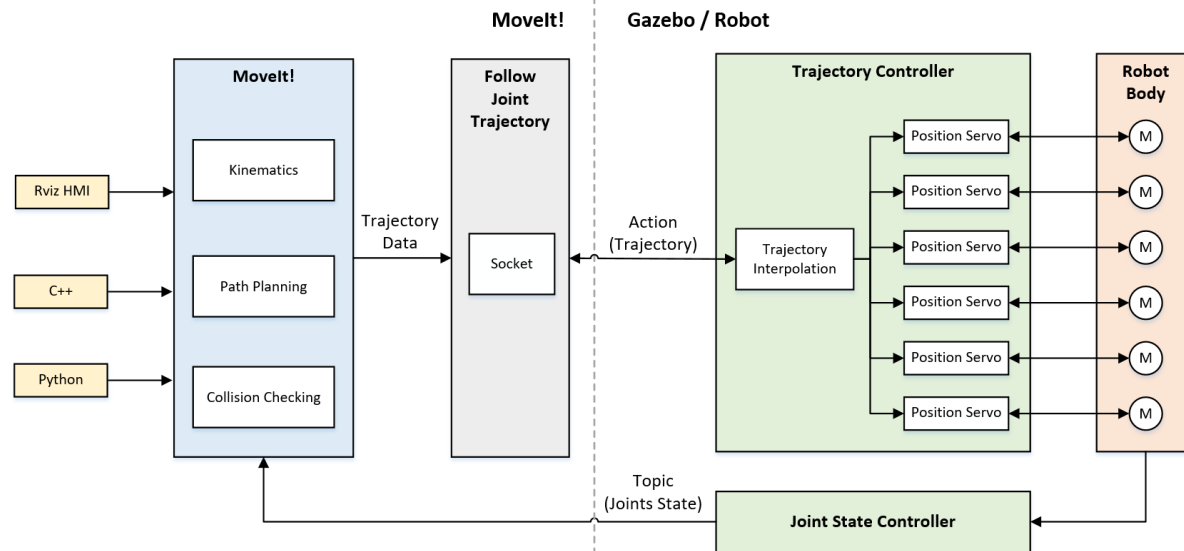


## ROS-I代码浅析

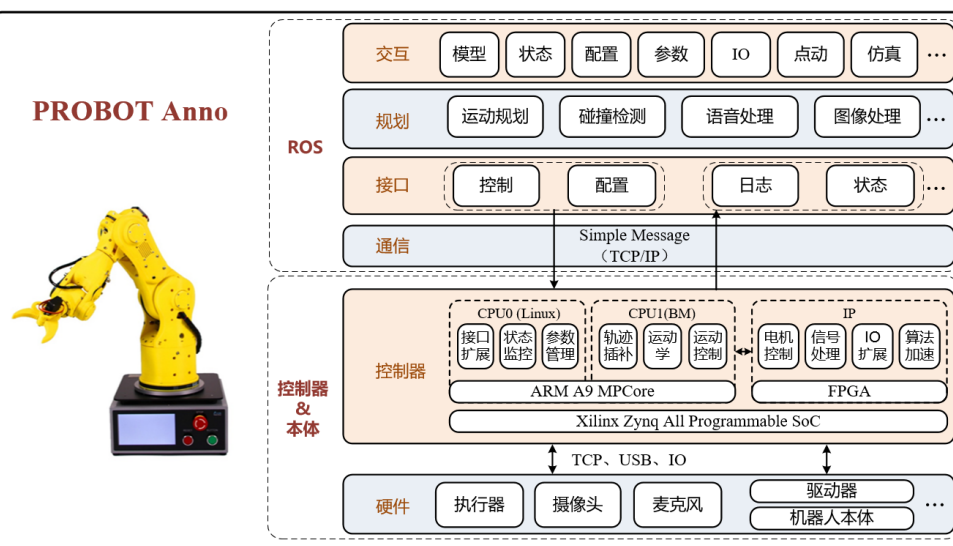


# 1. 课程总结 —— 11.基于ROS设计一款机械臂控制系统

## ROS控制系统 设计方法



## PROBOT Anno 控制系统案例分析





## 2. 进阶攻略



## 2. 进阶攻略



OpenManipulator



Baxter



Fetch



FRANKA EMIKA Panda



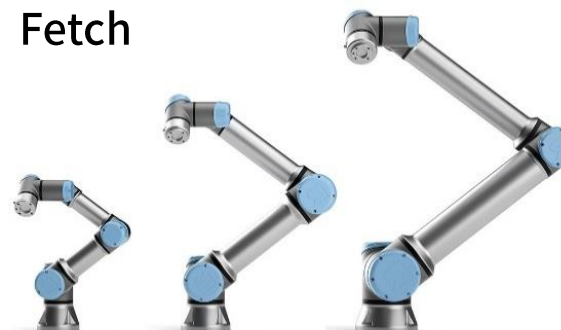
Kinova JACO



Turtlebot



Han's Cute



Universal Robots



PROBOT Anno

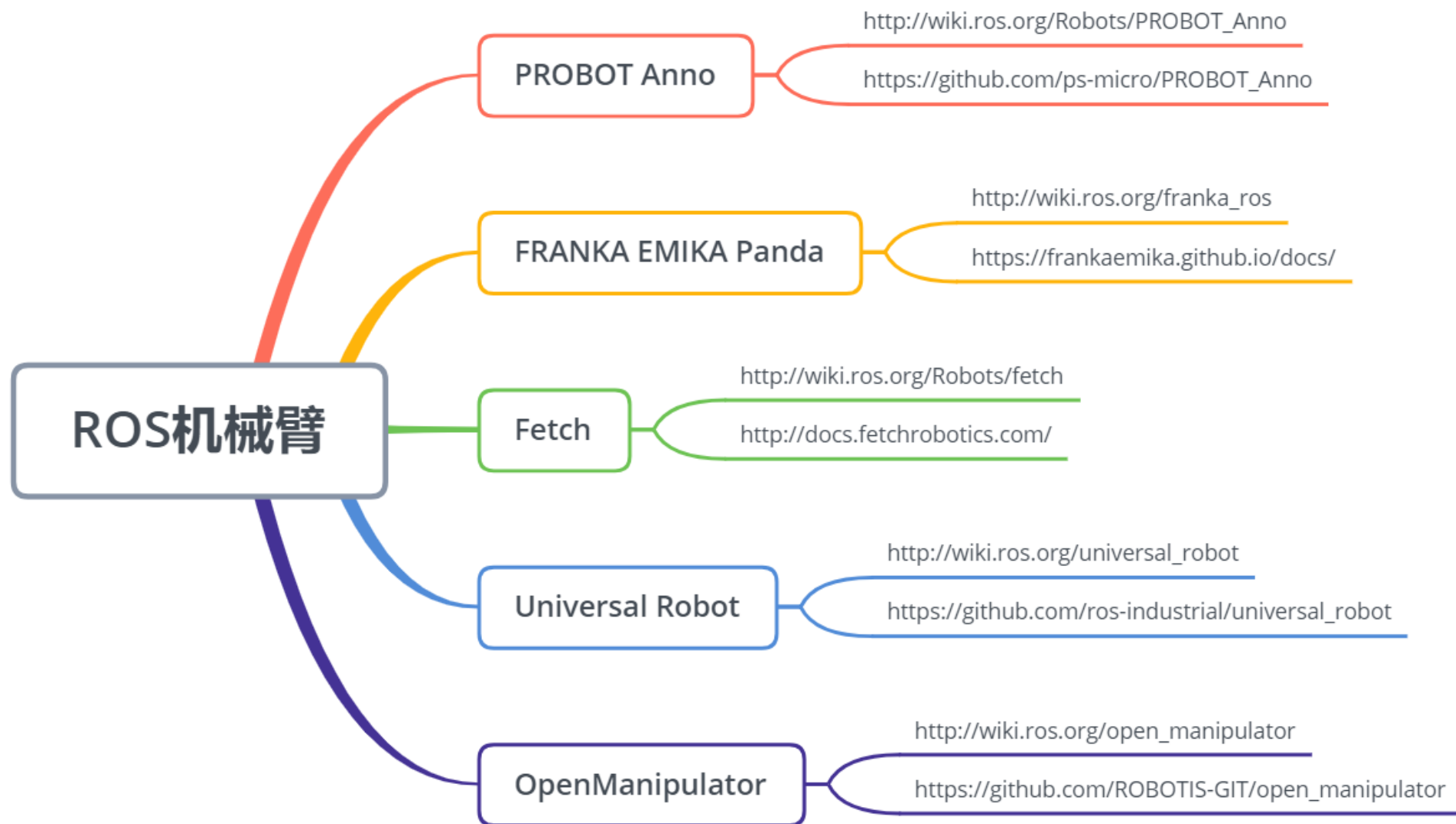


Clearpath Husky





## 2. 进阶攻略

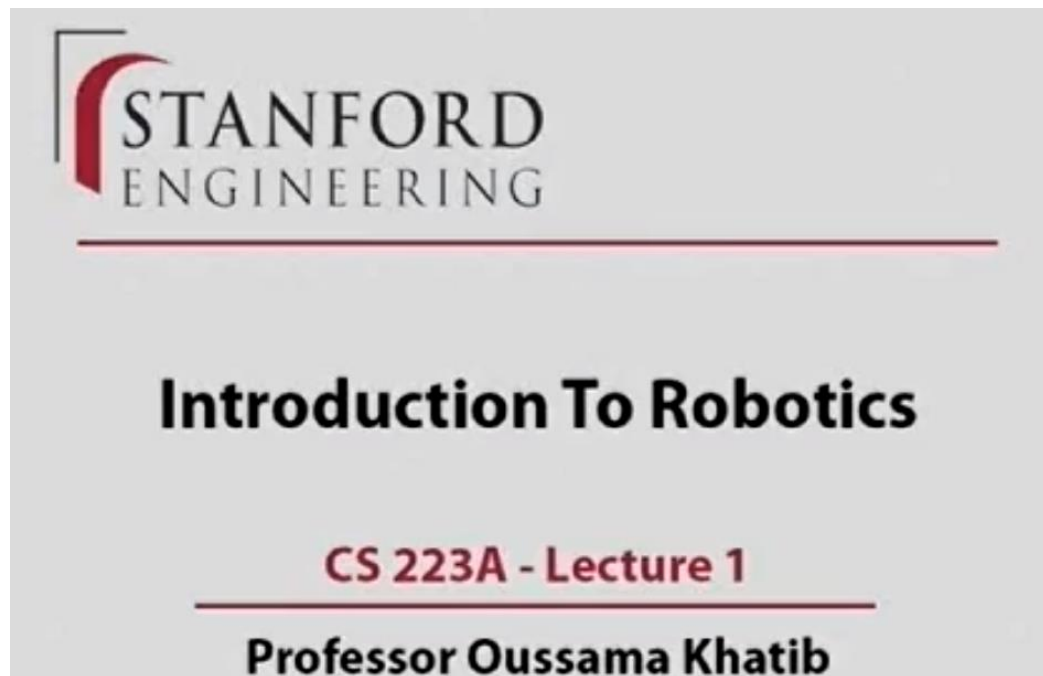




## 3. 资源整理



### 3. 资源整理



斯坦福大学公开课 —— 机器人学

<https://www.bilibili.com/video/av4506104/>

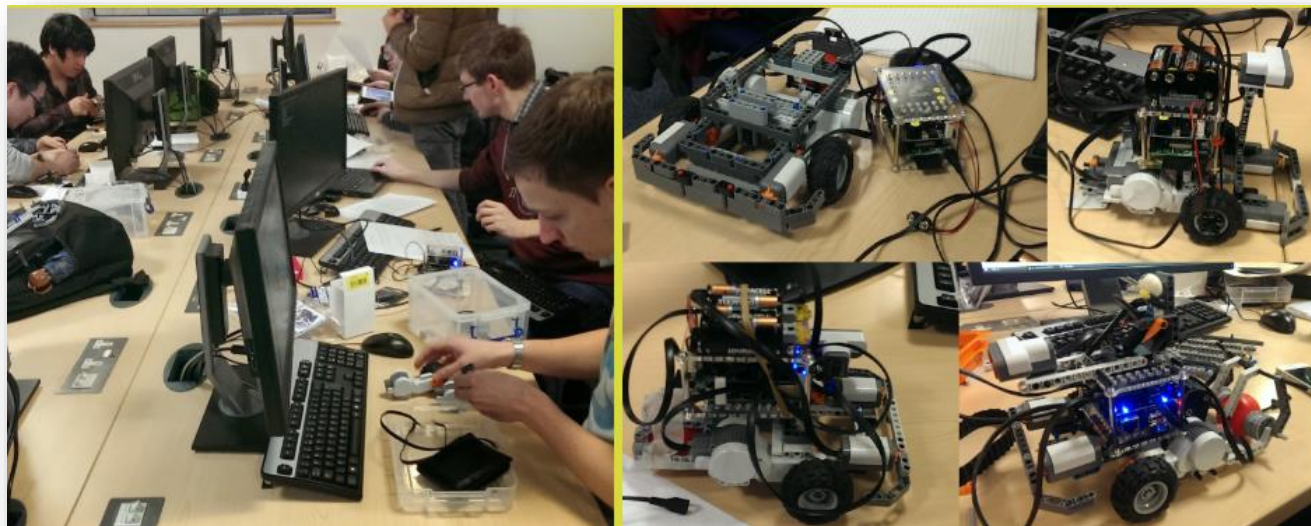


交通大学 —— 机器人学

<https://www.bilibili.com/video/av18516816/?p=2>



## 3. 资源整理



Andrew Davison的机器人学讲座课程  
<http://www.doc.ic.ac.uk/~ajd/Robotics/index.html>

## ETH - Robotic Systems Lab

<http://www.rsl.ethz.ch/education-students/lectures.html>

### Lectures

Number	Title of lecture	Teacher	Material
151-0851-00L	Robot Dynamics	M. Hutter, R. Siegwart, T. Stastny	<a href="#">Material</a> →
151-0662-00L	Programming for Robotics - Introduction to ROS	D. Jud, M. Wermelinger, Marko Bjelonic, P. Fankhauser, M. Hutter	<a href="#">Material</a> →



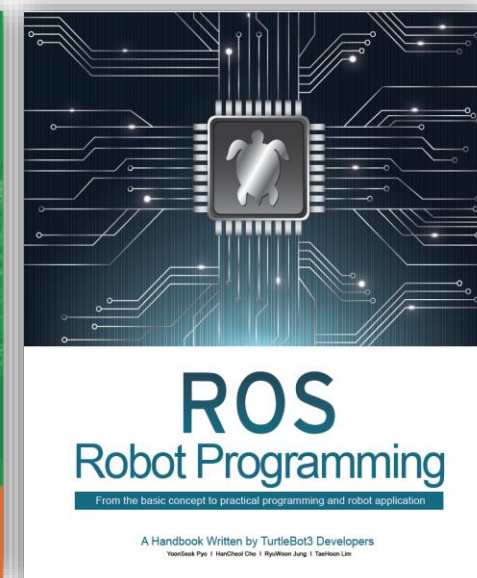
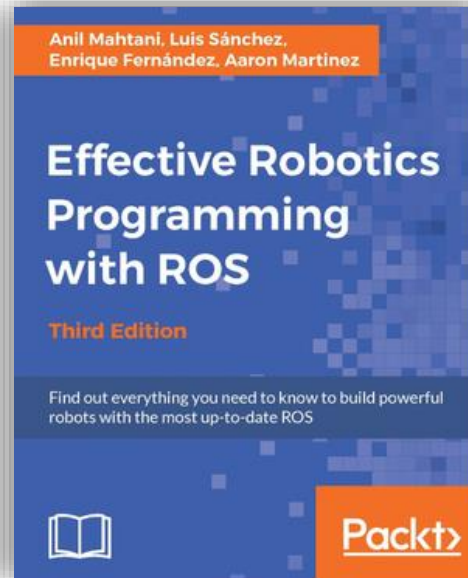
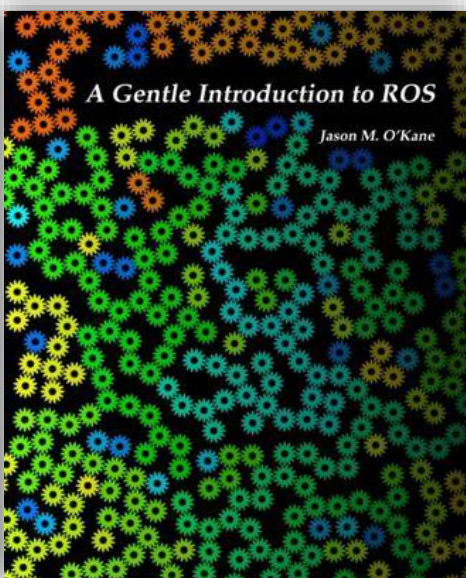
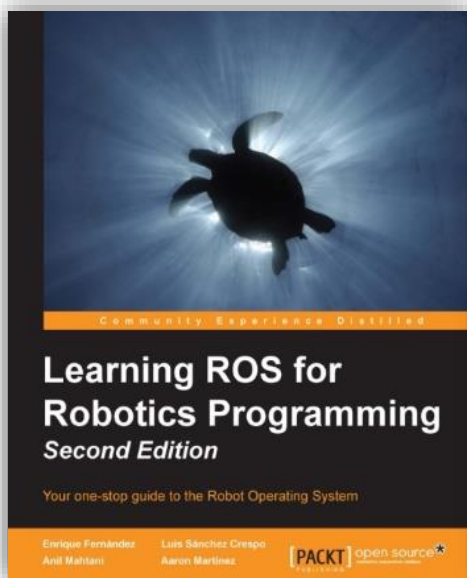
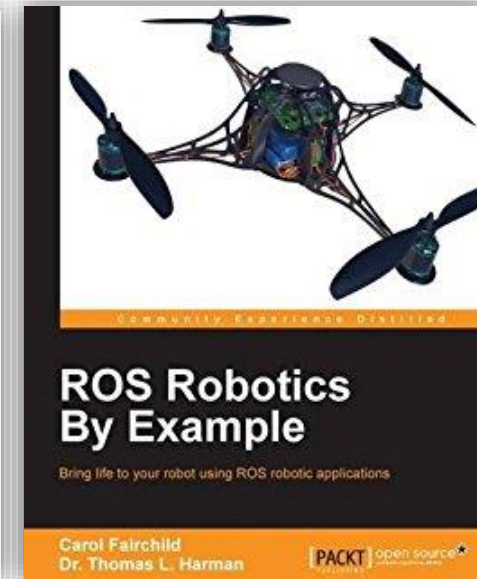
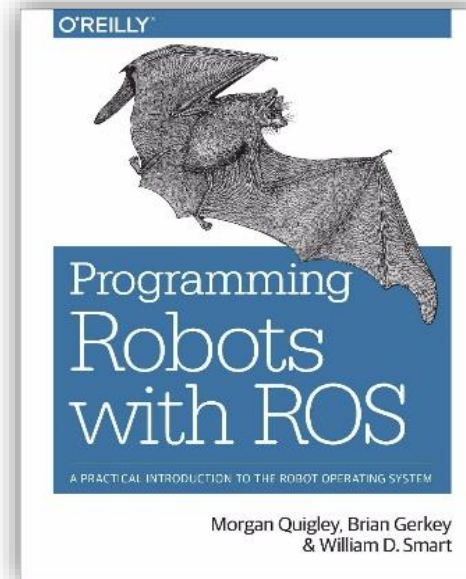
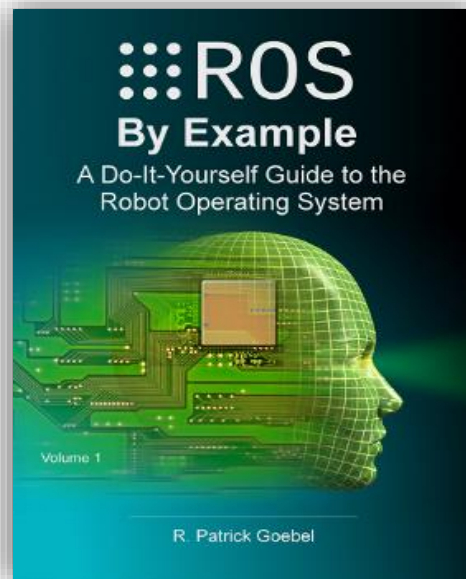
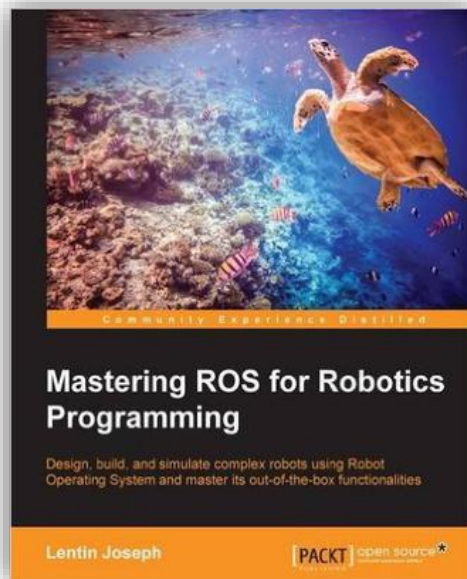
### 3. 资源整理

- ROS : <https://www.ros.org>
- ROS Wiki : <http://wiki.ros.org/>
- ROSCon 2012 ~ 2019 : <https://roscon.ros.org>
- ROS Robots : <https://robots.ros.org/>
- Ubuntu Wiki : <https://wiki.ubuntu.org.cn>
- 古月居 : <http://www.gyh.ai>
- zhangrelay的专栏 : <https://blog.csdn.net/ZhangRelay>
- 易科机器人实验室: <http://blog.exbot.net/>
- 开源机器人学学习指南: <https://github.com/qqfly/how-to-learn-robotics>





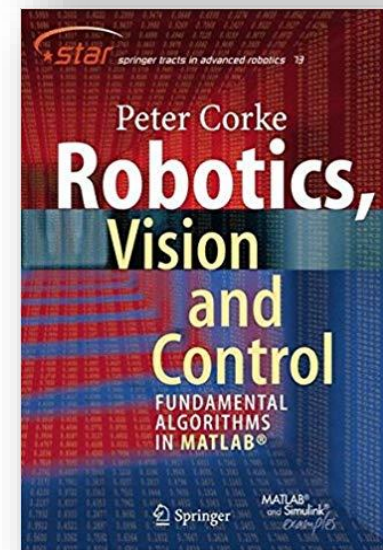
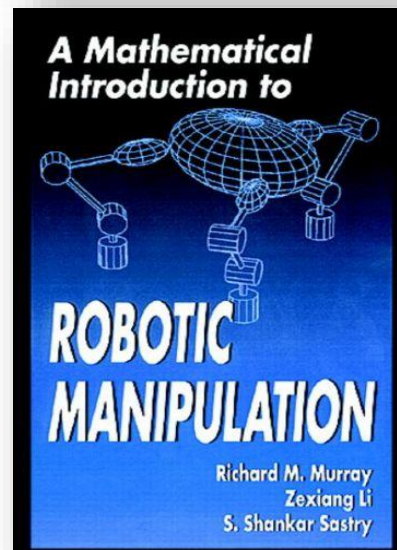
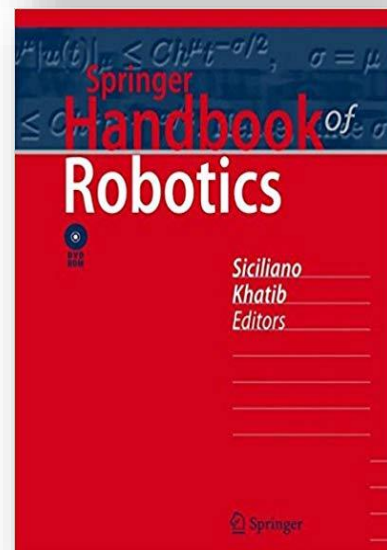
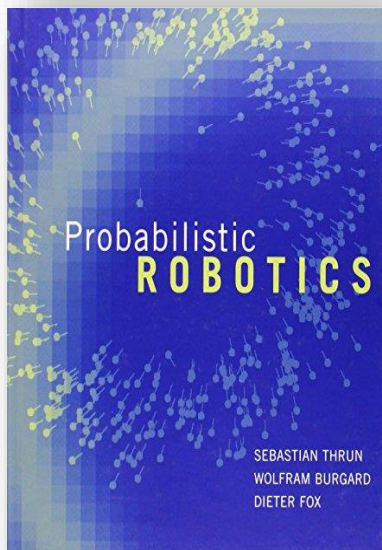
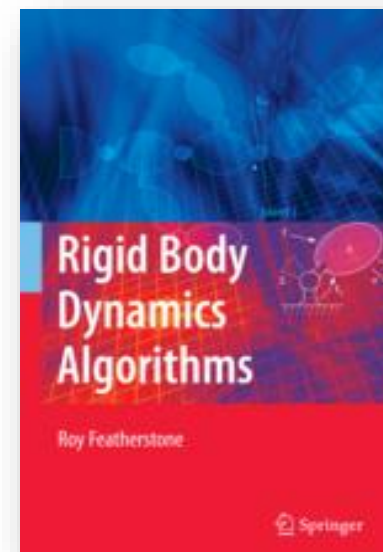
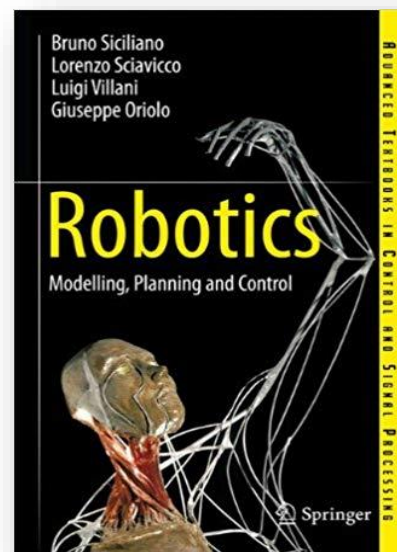
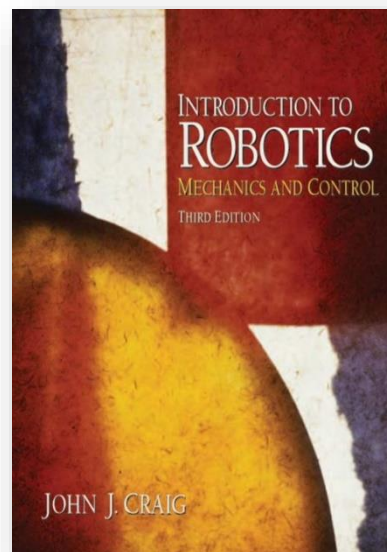
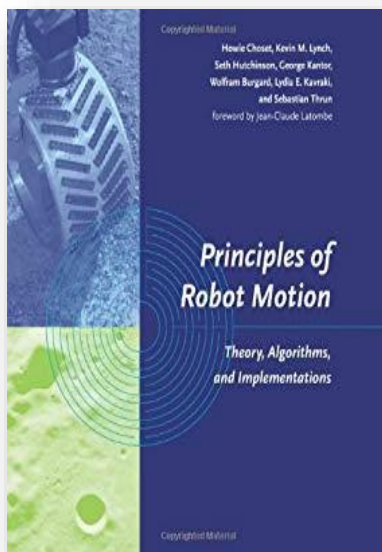
## 3. 资源整理







### 3. 资源整理





### 3. 资源整理

Google

 GitHub

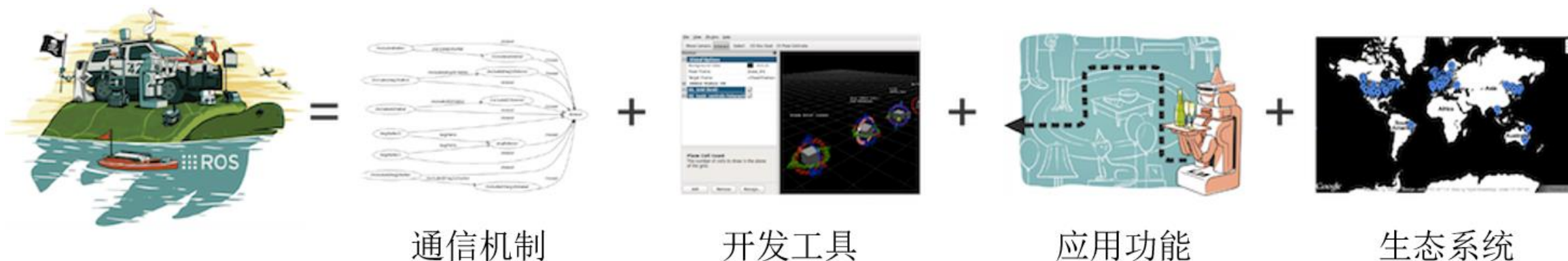
 YouTube

 Bilibili





### 3. 资源整理



**ROS是一个工具，也是一种生活方式，  
但他并不完美，也不是机器人开发的全部**

**What is ROS?** The name “robot operating system” is arguably a misnomer. Defining ROS succinctly is difficult, since it encompasses myriad aspects, including style of programming (notably, relying on loosely-coupled, distributed nodes); interface definitions and paradigms for communications among nodes; interface definitions for incorporation of libraries and packages; a collection of tools for visualization, debugging, data logging and system diagnostics; a repository of shared source code; and bridges to multiple useful, independent open-source libraries. **ROS is thus more of a way of life for robot programmers than simply an operating system.** Definitions of ROS are drawn from the following sources.

From the ROS wiki (<http://wiki.ros.org/ROS/Introduction>):



# 分享就是力量



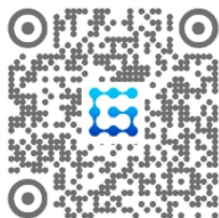
**怕什么真理无穷，进一寸有一寸的欢喜。**

**——胡适**

# Thank You

怕什么真理无穷，进一寸有一寸的欢喜

更多精彩，欢迎关注



古月居



古月春旭

古月居-GYH ♂  
UID: 386919562

