工程实践与科技创新 IV-E

Lecture 1: 绪论

授课老师: 张晗、王景川 *2023 年春季*

1 ROS 简介

1.1 机器人操作系统 ROS

1. ROS 是机器人操作系统（Robot Operating System）的英文缩写，是一个适用于机器人的开源的元操作系统。

2. 它提供了操作系统应有的服务，包括硬件抽象，底层设备控制，常用函数的实现，进程间消息传递，以及包管理。它也提供用于获取、编译、编写和跨计算机运行代码所需的工具和库函数。

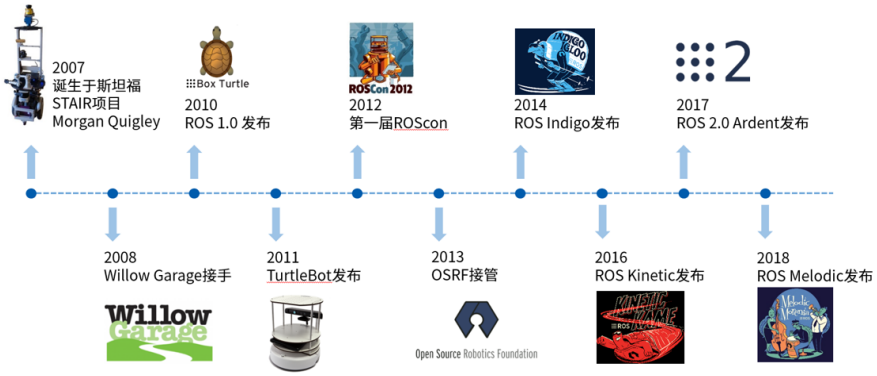

Figure 1: ROS 系统发展进程

1.2 ROS 的架构

1. ROS 就是一套通信机制、一套开发工具、一系列应用工具加一个庞大的生态系统组成的集合，目标是提高机器人研发中的软件复用率。针对这样的目标，如图 Fig. 2，ROS 有五大特点。

2. ROS 的组成有以下四大部分，即通讯机制 + 开发工具 + 应用功能 + 生态系统。

3. ROS 提供了很多开发工具，目的是提高机器人开发效率。第一个是命令行工具，可以直接在 terminal 内进行操作；还有 TF 工具，可以帮助进行坐标变换；还有 QT 工具箱，提供很多可视化的工具；RVIZ 是一个三维可视化工具，可以显示机器人的全部数据；Gazebo 是一个三维仿真平台，可以进行机器人仿真。

4. 如图 Fig. 4，ROS 在生态层面讲就是这么几个重要的资源。发行版指 ROS 发布的版本。软件源中放置的是可直接安装的功能包。ROS Wiki 是 ROS 信息记录最完善的网站。邮件列表

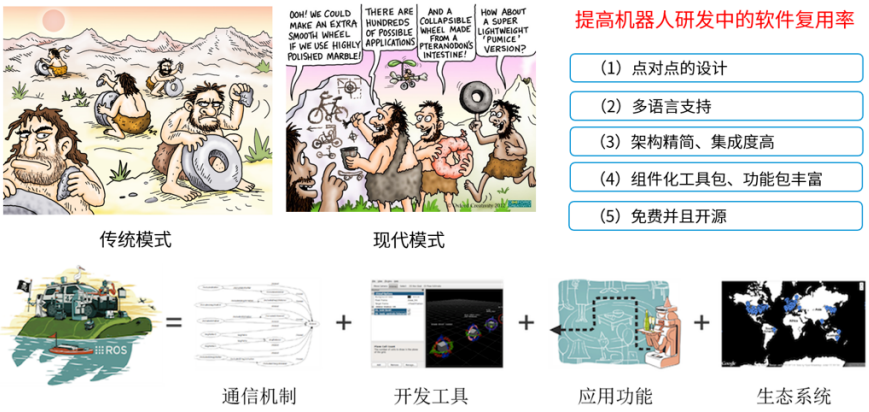

Figure 2: ROS 系统架构

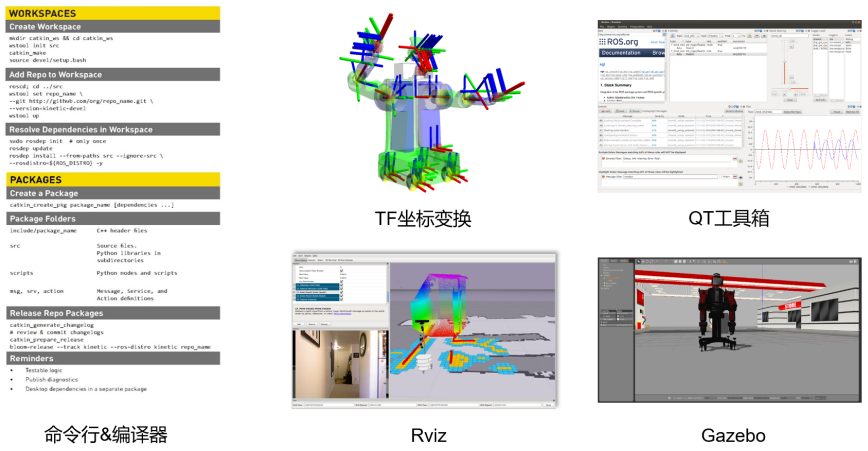

Figure 3: ROS 开发工具

更多的用在 ROS 更新维护，开发者使用较少。ROS Answers 是专门的 ROS 问答网站。博客会看到一些 ROS 的新闻、图片、视频等。

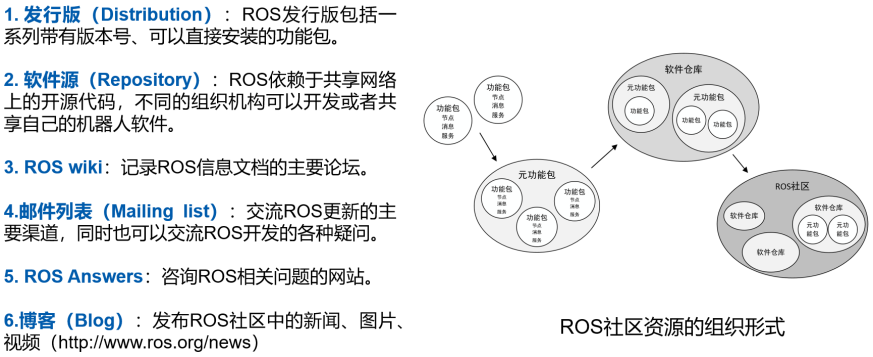

Figure 4: ROS 生态资源

1.3 ROS 的通讯机制

1. ROS 的通讯机制为松耦合分布式通讯。类似于图 Fig. 5在两个电脑中，分别有一个节点在使用 topic 进行通讯，为了更好的通讯，Central Node 负责去管理各个节点之间的通讯。

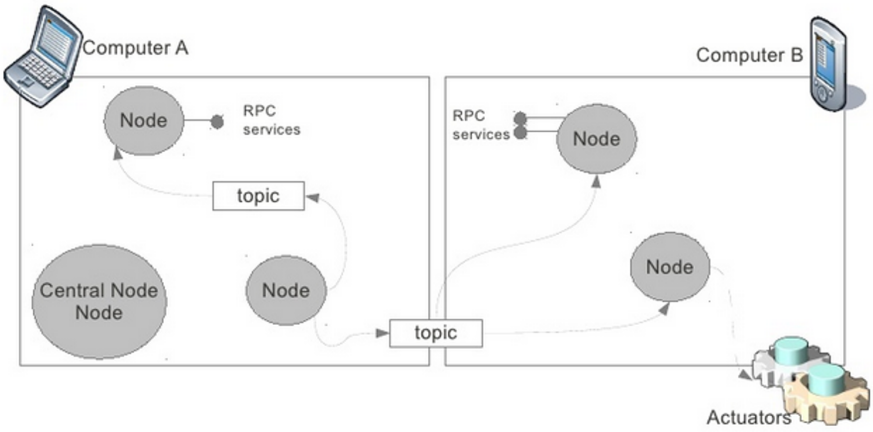

Figure 5: PC 通讯机制

2. ROS 通讯机制方面会看到这样一张图 (如图 Fig. 6)，机器人之间的通讯会形成一个网络，称为计算图。在图当中会有很多节点，椭圆部分，方框是节点的命名空间，箭头是数据传输的关系。这就是 ROS 在通讯层面的松耦合分布式通讯。松耦合就是说每一个节点不是和另一个节点强绑定的；分布式是指各个节点可以运行在不同的计算机内

3. 在 Ros 中，节点是最小可执行单元，就是每一个进程。编译完成后的每一个可在执行文件就是一个节点，实现一个功能。注意节点不能重名。节点管理器负责控制节点的注册及生成，他会跟踪和记录通讯，会辅助节点查找，建立连接，也会提供全局的参数服务器。

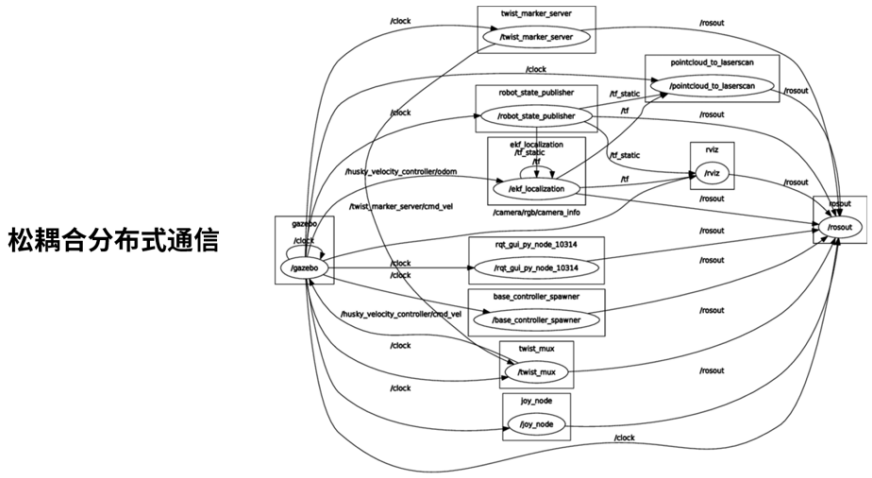

Figure 6: ROS 通讯机制

4. 话题是节点之间传输数据的一种单向的异步通讯的机制。下面这张图发布者通过话题发布数据给订阅者，单向的传输。其中通讯时的数据定义即为消息，具有一定的类型和数据结构，可以使用 ROS 提供的，也可以使用自定义的消息类型。

1.4 ROS 的模块化设计理念

在我们对 ROS 系统有了一个初步的了解后，我们需要搞清楚一些事，ROS 为什么会设计成这几个模块？它的通讯机制为什么会是这种方式？

1.4.1 模块化设计

首先最基础的，ROS 的模块化设计有以下优点：

1. 便于同时开发各个模块，提高协作开发效率
2. 模块内容可以更换、扩展，便于实现更复杂的功能
3. 模块具有复用性，便于快速搭建系统

1.4.2 一个例子

以移动机器人系统为例，我们来说明通讯机制的需求。通常情况下，移动机器人由传感器系统，底层运动系统，上位机系统，人机交互系统组成，各部分又由对应的软硬件搭建。针对一次具体的任务，比如将机器人导航到指定的位姿，一般可以分为以下过程：

1. 自定位，根据传感器采集的环境数据得到机器人空间坐标
2. 目标位姿设定后进行路径规划
3. 机器人沿着规划的路径移动
4. 满足设定的停止条件后停止移动

在考虑其通信时，存在以下问题

1. 上位机如何收到传感器的数据？如何向底层运动系统发出控制信号？
2. 多个传感器同时以不同频率发送数据时上位机如何接收？
3. 如何在接受传感器数据的同时发送控制信号？
4. 针对不同传感器发送的不同类型的数据如何处理？
5. 如果有多个移动机器人如何协调？

可以看到，ROS 的通讯机制设计较好的解决了这些问题，进而可以在使用 ROS 系统的前提下有效完成对应的任务。

2 Gazebo 简介

Figure 7: 机器人仿真软件 Gazebo

Gazebo 软件是一个机器人仿真工具，可以用于快速验证算法、设计机器人、验证回归测试等。gazebo 能够准确有效地模拟室内室外中的机器人种群，使强大的物理引擎，高质量的图形以及方便的编程和图形化接口触手可及。同时，gazebo 目前免费并且拥有一个活跃的社区，对于入门者十分友好。Gazebo 是课程的主要仿真平台，我们会在 gazebo 中搭建实验的环境模型、机器人模型与噪声模型等。

2.1 构建机器人运动仿真模型

在 Gazebo 里，提供了最基础的三个物体，球体，圆柱体，立方体，利用这三个物体以及它们的伸缩变换或者旋转变换，可以设计一个最简单的机器人三维仿真模型。同时，Gazebo 提供了机器人的运动仿真，通过 Model Editor 下的 plugin，来添加我们需要验证的算法文件，就可以在 Gazebo 里对机器人的运动进行仿真

2.2 构建现实世界各种场景的仿真模型

Gazebo 可以建立一个用来测试机器人的仿真场景，通过添加物体库，放入垃圾箱，雪糕桶，甚至是人偶等物体来模仿现实世界，还可以通过 Building Editor，添加 2D 的房屋设计图，在设计图基础上构建出 3D 的房屋

2.3 构建传感器仿真模型

Gazebo 拥有一个很强大的传感器模型库，包括 camera，depth camera，laser，imu 等机器人常用的传感器，并且已经有模拟库，已经可以直接使用，也可以自己从零开始创建一个新的传感器，添加它的具体参数，甚至还可以添加传感器噪声模型，让传感器更加真实。

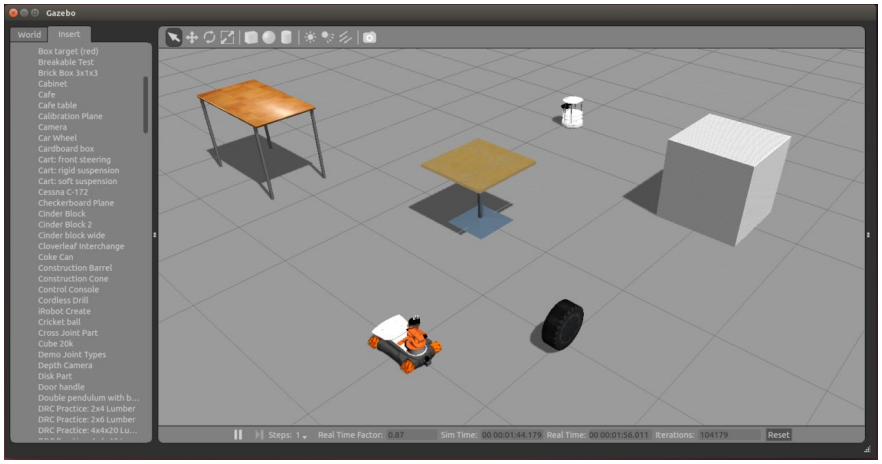

Figure 8: Gazebo 仿真页面

2.4 为机器人模型添加现实世界的物理性质

Gazebo 里有 force，physics 的选项，可以为机器人添加例如重力，阻力等，Gazebo 有一个很接近真实的物理仿真引擎，但有一点需要注意，一般的地面是没有阻力的，和现实世界有区别

3 ROS 演示 *

3.1 小海龟仿真器

安装完成后可以使用 ROS 自带的小海龟例程进行验证。首先打开一个终端运行 roscore，启动 rosmaster；再打开一个终端启动小海龟仿真器，最后再打开一个终端启动海龟控制节点。此时可以使用方向键控制小海龟运动即可说明 ROS 安装成功。注意控制小海龟时要注意输入焦点在控制节点对应的终端中。

1. roscore 指令启动：

   roscore

   roscore 是基于 ROS 的系统的先决条件的节点和程序的集合，必须运行 roscore 才能使 ROS 节点进行通信。

2. 启动小海龟节点：

   rosrun turtlesim turtlesim_node

   启动成功后会打开一个可视化终端，效果如图 Fig. 9。注意：其中小海龟的图样是随机的，会有所不同。

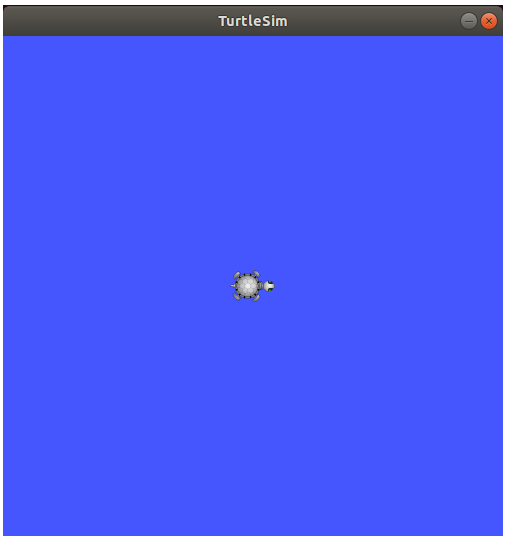

Figure 9: turtlesim_node

3. 启动小海龟键盘控制节点：

   rosrun turtlesim turtle_teleop_key

   如图图 Fig.10，打开键盘节点后，使用键盘的方向键即可控制小海龟向对应方向运动。注意，要保证输入焦点在 turtle_teleop_key 的终端才能正常使用。

3.2 查看 ROS 计算图

1. rqt_graph 提供了一个 GUI 插件，用于可视化 ROS 计算图。可以使用 rqt_graph 命令打开可视化的计算图：

   rqt_graph

2. 打开后即可看到如图 Fig.11所示的可视化弹窗。其中显示了当前所开启的全部节点，包括 turtlesium 和 turtle_teleop_key 及其对应的话题。

3.3 ROS 的常用命令

rosnode 是一个命令行工具，用于显示有关 ROS 节点的调试信息，包括发布，订阅和连接。

1. 其中 rosnode info 可以显示节点的相关信息，包括发布和订阅:

```
$ rosnode info /turtlesim
```

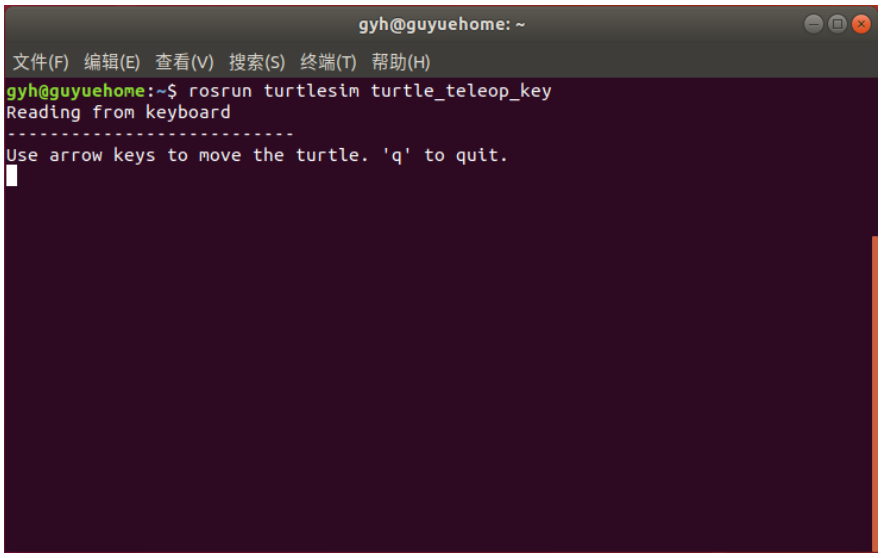

Figure 10: turtle_teleop_key

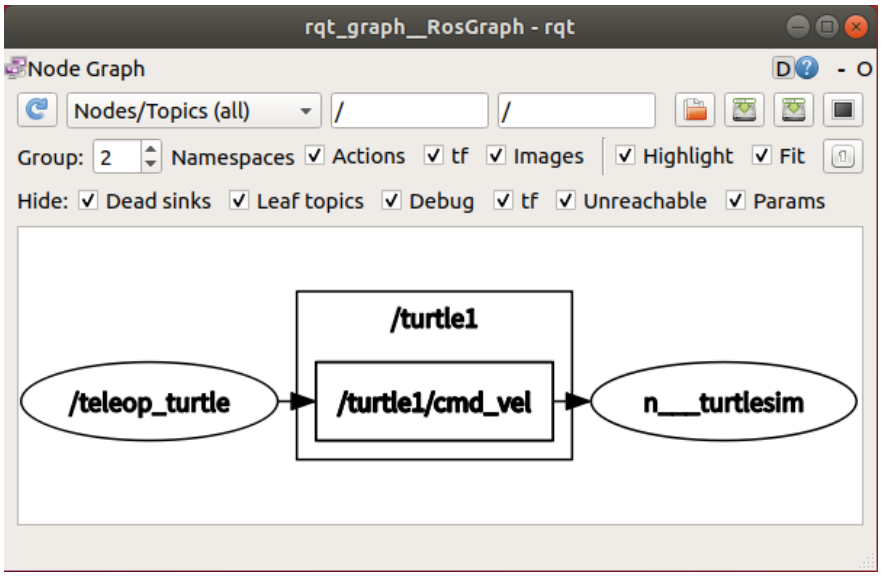

Figure 11: rqt_gragh

```
Node [/turtlesim]
Publications:
 * /rosout [rosgraph_msgs/Log]
 * /turtle1/color_sensor [turtlesim/Color]
 * /turtle1/pose [turtlesim/Pose]
Subscriptions:
 * /turtle1/cmd_vel [unknown type]
Services:
 * /clear
 * /kill
 * /reset
 * /spawn
 * /turtle1/set_pen
 * /turtle1/teleport_absolute
 * /turtle1/teleport_relative
 * /turtlesim/get_loggers
 * /turtlesim/set_logger_level
contacting node http://192.168.3.154:45825/ ...
Pid: 2406
Connections:
 * topic: /rosout
    * to: /rosout
    * direction: outbound (36671 - 192.168.3.154:35814) [24]
    * transport: TCPROS
```

2. rosnode list 可以显示当前节点的列表:

```
$ rosnode list
/rosout
/teleop_turtle
/turtlesim
```

3. rostopic 包含 rostopic 命令行工具，用于显示有关 ROS 话题的调试信息，包括发布者，订阅者，发布率和 ROS 消息。rostopic list 可以显示当前话题列表:

```
$ rostopic list
/rosout
/rosout_agg
/turtle1/cmd_cel
/turtle1/color_sensor
/turtle1/pose
```

4. rostopic info 可以打印话题有关的信息:

```
$ rostopic info /turtle1/cmd_vel
Type: geometry_msgs/Twist
```

```
Publishers:
 * /teleop_turtle (http://192.168.3.224:42663/)

Subscribers:
 * /turtlesim (http://192.168.3.224:40187/)
```

5. rostopic echo 可以显示话题发布的消息。例如查看小海龟的位姿数据:

```
$ rostopic echo /turtle1/pose
x: 5.544444561
y: 5.544444561
theta: 0.0
linear_velocity: 0.0
angular_velocity: 0.0
```

6. rostopic pub 可以发布话题数据，可以通过此命令发布小海龟的速度消息，使小海龟进行圆周运动:

```
$ rostopic pub /turtle1/cmd_vel geometry_msgs/Twist
"linear:
x:10.0
y:0.0
z:0.0
angular:
x:0.0
y:0.0
z:10.0"
```

其中可以增加 -r 参数，表示速率，默认为 10hz。

rosservice 包含用于列出和查询 ROS 服务的命令行工具，它包含一个 Python 库，用于检索有关服务的信息并动态调用它们。

7. rosservice list 显示活动中的服务的信息:

```
$ rosservice list
/clear
/kill
/reset
/rosout/get_loggers
/rosout/set_logger_level
/spawn
/teleop_turtle/get_loggers
/teleop_turtle/set_logger_level
/turtle1/set_pen
/turtle1/teleport_absolute
/turtle1/teleport_relative
/turtlesim/get_loggers
/turtlesim/set_logger_level
```

8. rosservice info 可以打印服务有关的信息：

```
$ rosservice info /turtle1/set_pen
Node: /turtlesim
URI: rosrpc://nvidia:35261
Type: turtlesim/SetPen
Args: r g b width off
```

9. rosservice call 可以用输入的参数调用服务：

```
$ rosservice call /turtle1/set_pen "{r:255,g:0,b:0,width:5,'off':0}"
```

课堂示例是请求/turtle1/set_pen 服务的命令。所使用的“255 0 0 5 0”是对应于用于/turtle1/set_pen 服务的参数 (r，g，b，width，off) 的值。红色的 r 的最大值是 255，因为 g 和 b 都是 0，所以笔的颜色是红色的。width 设置为 5，off 为 0（假）。