Report on Robot Operating System (ROS)

Yanyu Zhang

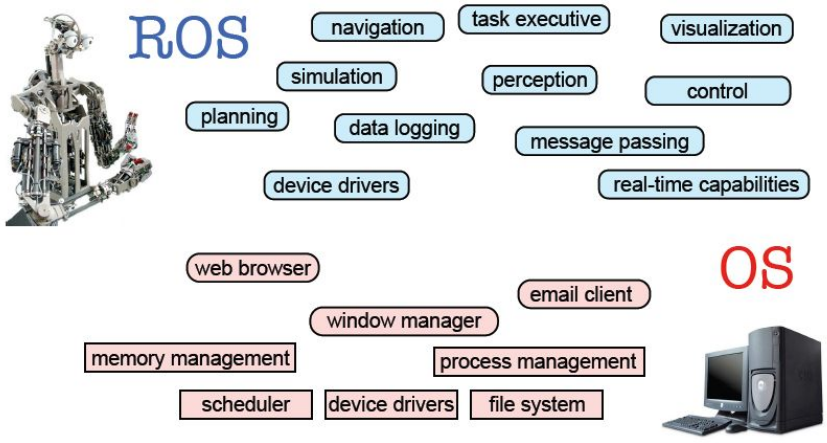
A. The Introduction and Features of ROS

a. What is the ROS

The Robot Operating System (ROS) is a flexible framework for writing robot software. It is a collection of tools, libraries, and conventions that aim to simplify the task of creating complex and robust robot behavior across a wide variety of robotic platforms.[1]

b. Features of ROS

1. ROS is a free and open-source robot operating system.
2. A set of software libraries and tools that help you build robot applications that work across a wide variety of robotic platforms. such as visualization, logging, plotting data streams, etc.
3. A ROS system is comprised of a number of independent nodes, each of which communicates with the other nodes using a publish / subscribe messaging model.
4. Peer to Peer: ROS systems consist of numerous small computer programs which connect to each other and continuously exchange messages.
5. Multi-Lingual: ROS software modules can be written in any language for which a client library has been written. Currently client libraries exist for C++, Python, LISP, Java, JavaScript, MATLAB, Ruby, and more. [2]



c. ROS has two "sides"

1. The operating system side, which provides standard operating system services such as:
   hardware abstraction;
   low-level device control;
   implementation of commonly used functionality;
   message-passing between processes;
   package management

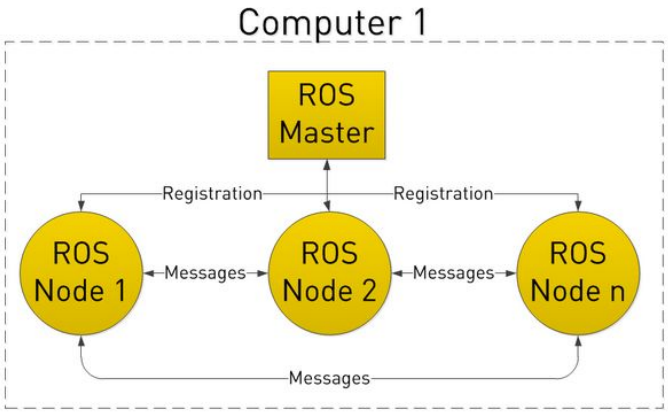
2. A suite of user contributed packages that implement common robot functionality such as planning, perception, vision, manipulation, etc. [3]

d. ROS support many robots, such as pioneer 3 and Husky [4]

B. ROS Main Core Concepts

a. Master and Node Concept

1. The Master allows all other ROS pieces of software (Nodes) to find and talk to each other. Master is the core node of ROS, called roscore.
2. Nodes are publishing and subscribing to different Topics.
3. Master stores topics and services registration information for ROS nodes. Nodes then establish connections as appropriate.
4. Also makes callbacks to nodes when registration information changes.
5. Allows nodes to dynamically create connections as new nodes are run.

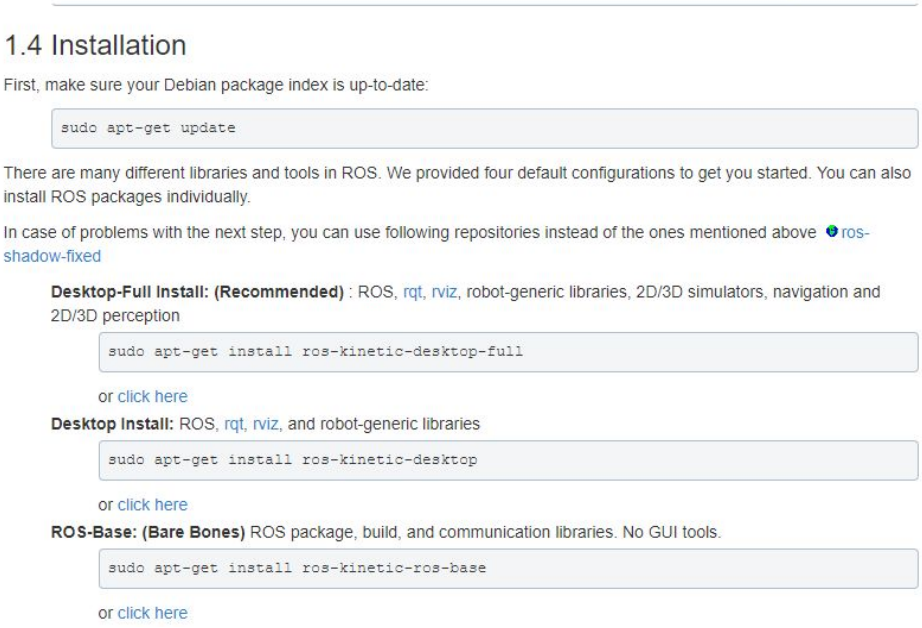


b. Master Node

1. Provides connection information to nodes so that they can transmit messages to each other.
2. Every node connects to a master at startup to register details of the message streams they publish, and the streams to which they subscribe.
3. When a new node appears, the master provides it with the information that it needs to form a direct peer-to-peer connection with other nodes publishing and subscribing to the same message topics.

C. How to install and run ROS in Ubuntu

a. ROS can run in the Ubuntu environment, the install step as following link: http://wiki.ros.org/kinetic/Installation/Ubuntu



1.4 Installation

First, make sure your Debian package index is up-to-date:

```
sudo apt-get update
```

There are many different libraries and tools in ROS. We provided four default configurations to get you started. You can also install ROS packages individually.

In case of problems with the next step, you can use following repositories instead of the ones mentioned above ros-shadow-fixed

Desktop-Full Install: (Recommended) : ROS, rqt, rviz, robot-generic libraries, 2D/3D simulators, navigation and 2D/3D perception

```
sudo apt-get install ros-kinetic-desktop-full
```

or click here

Desktop Install: ROS, rqt, rviz, and robot-generic libraries

```
sudo apt-get install ros-kinetic-desktop
```

or click here

ROS-Base: (Bare Bones) ROS package, build, and communication libraries. No GUI tools.

```
sudo apt-get install ros-kinetic-ros-base
```

or click here

b. The basic ROS commands are as follows:

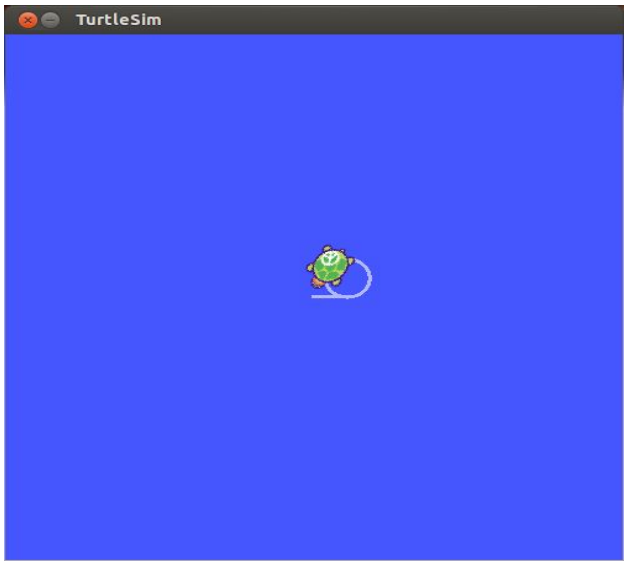
Common Command-Line Tools	
roscore	A collection of nodes and programs that are pre-requisites of a ROS-based system. You must have a roscore running in order for ROS nodes to communicate. http://wiki.ros.org/roscore Usage: `roscore`
rosmsg	The rosmsg command-line tool displays information about ROS message types. http://wiki.ros.org/rosmsg Usage: `rosmsg [options]`
rossrv	The rossrv command-line tool displays information about ROS services. http://wiki.ros.org/rosmsg Usage: `rossrv [options]`
rosrun	The rosrun allows you to run an executable in an arbitrary package from anywhere without having to give its full path. http://wiki.ros.org/rosbash Usage: `rosrun package executable`
rosnode	Displays debugging information about ROS nodes, including publications, subscriptions and connections. http://wiki.ros.org/rosnode Usage: `rosnode [options]`
roslaunch	Starts ROS nodes locally and remotely via SSH, as well as setting parameters on the parameter server. http://wiki.ros.org/roslaunch Usage: `roslaunch [options]`
rostopic	A tool for displaying debug information about ROS topics, including publishers, subscribers, publishing rate, and messages. http://wiki.ros.org/rostopic Usage: `rostopic [options]`
rosparam	A tool for getting and setting ROS parameters on the parameter server using YAML-encoded files. http://wiki.ros.org/rosparam Usage: `rosparam [options]`
rosservice	A tool for listing and querying ROS services. http://wiki.ros.org/rosservice Usage: `rosservice [options]`

c. Test the turtlesim package in the ROS

>> roscore

>> rosrun turtlesim turtlesim_node

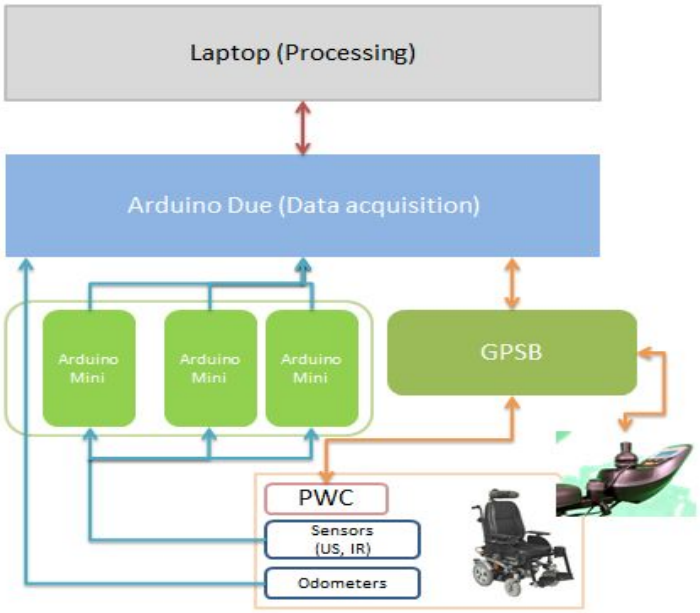
>> rosrun turtlesim turtle_teleop_key



D. Self-driving wheelchair based on ROS

a. In the medical sector, and mainly for dependent patients with movement disabilities, controlling an electric powered wheelchair could prove a challenging task. Thus, implementing an autonomous navigation algorithm for static/dynamic environments

could provide an easier way to move. The work focuses on integrating the geolocalization algorithm within the ROS framework. [5]



References

[1] Quigley, Morgan, Ken Conley, Brian Gerkey, Josh Faust, Tully Foote, Jeremy Leibs, Rob Wheeler, and Andrew Y. Ng. "ROS: an open-source Robot Operating System." In *ICRA workshop on open source software*, vol. 3, no. 3.2, p. 5. 2009.

[2] Juan, Sergi Hernandez, and Fernando Herrero Cotarelo. "Multi-master ros systems." *Institut de Robotics and Industrial Informatics* (2015): 1-18.

[3] Y. Zhang, X. Wang, X. Wu, W. Zhang, M. Jiang and M. Al-Khassaweneh, "Intelligent Hotel ROS-based Service Robot," *2019 IEEE International Conference on Electro Information Technology (EIT)*, Brookings, SD, USA, 2019, pp. 399-403.

[4] R. Mishra and A. Javed, "ROS based service robot platform," *2018 4th International Conference on Control, Automation and Robotics (ICCAR)*, Auckland, 2018, pp. 55-59.

[5] Nasri, Yassine, Vincent Vauchey, Redouane Khemmar, Nicolas Ragot, Konstantinos Sirlantzis, and Jean-Yves Ertaud. "Ros-based autonomous navigation wheelchair using omnidirectional sensor." *International Journal of Computer Applications* 133, no. 6 (2016): 12-17.