Module 3 - Nodes and Topics

ME4140 - ROS Workshop

Mechanical Engineering

Tennessee Technological University

Lecture 3 - Nodes and Topics

Lecture 3 - Nodes and Topics

- Components of a ROS system
- Communication and Message Passing
- Software Packages
- Tutorial 3 - Turtlesim

Components of a ROS system

Master Node

- *The ROS Master provides naming and registration services to the rest of the nodes in the ROS system.***
- master node runs first

```
roscore
```

- for distributed computing use single master

```
ROS_MASTER_URI=http://12345
```

Components of a ROS system

Nodes

- *A node is a process that performs computation.***
- each 'program' or 'element' of the robot is a node. Examples:
  - sensor
  - hardware driver
  - navigation
  - keyboard or joystick
- start or run node individually after master

```
rosrun <packagename> <nodename> <options>
```

Communication and Message Passing

Nodes can communicate in three different ways.

- topics - publishing and subscribing

```
rostopic
```

- parameter server - static data

```
rosparam
```

- services - subroutine call (RPC)

```
rosservice
```

Communication and Message Passing

Topics: ROS nodes can communicate by **publishing** and **subscribing** to topics. A topic is information generated by a publishing node that is made available to a subscribing node or nodes in the ROS system.

- A node can publish a topic. This node is a publisher.
- A node can subscribe to a topic. This node is a subscriber.
- Most nodes publish and subscribe to multiple topics.
- Integrate built-in ROS nodes and modify your own ROS nodes in C++, Python, and even MATLAB

Software Packages

In general, software is organized in packages

- Definition: A suite of programs that function as a single entity to accomplish a task, or group of related tasks.
- *Software in ROS is organized in packages. A package might contain ROS nodes, a ROS-independent library, a dataset, configuration files, a third-party piece of software, or anything else that logically constitutes a useful module.***
- a collection of related nodes, each node belongs to a package
- many pre-built packages with ros installation: **-desktop-full**

Software Packages

Installing Packages in Linux with package manager **apt**

- update Ubuntu before installing software packages

```
sudo apt update
```

- install pre-built packages from repository

```
sudo apt install ros-<distribution>-<packagename>
```

- remove installed packages from repository

```
sudo apt remove ros-<distribution>-<packagename>
```

Tutorial 3 - Turtlesim

- **Overview:** Turtlesim is a simple ROS robot! Your exercise is to install and test a simple simulator called turtlesim.
- **Assignment:** Complete the tutorial *Tutorial3 Turtlesim Testdrive*. You must be able to drive your turtle around the screen using the keyboard.
- **Deliverable:** Write a one to two paragraph summary of what you accomplished and what you struggled with the most. Include an image of the turtlesim window after you have driven a pattern.
- **Next Week:** After completion of Module 3, you are almost ready for a better robot. Next, you will learn to customize your ROS environment and build a custom package and C++ node.