

# ROS Tutorial

EECS 567/398







11.13.2017

# What is ROS

- ROS is a flexible framework for writing robot software
- ROS provides tools, libraries, and conventions such that different developers can share and collaborate easily

# ROS Installation

- [Install ROS Indigo](#) (On supports ubuntu 13.10 and 14.04)

Distro	Release date	Poster	Turtle, turtle in tutorial	EOL date
ROS Melodic Morenia	May, 2018	TBD	TBD	May, 2023
ROS Lunar Loggerhead	May 23rd, 2017			May, 2019
ROS Kinetic Kame (Recommended)	May 23rd, 2016			April, 2021 (Xenial EOL)
ROS Jade Turtle	May 23rd, 2015			May, 2017
ROS Indigo Igloo	July 22nd, 2014			April, 2019 (Trusty EOL)

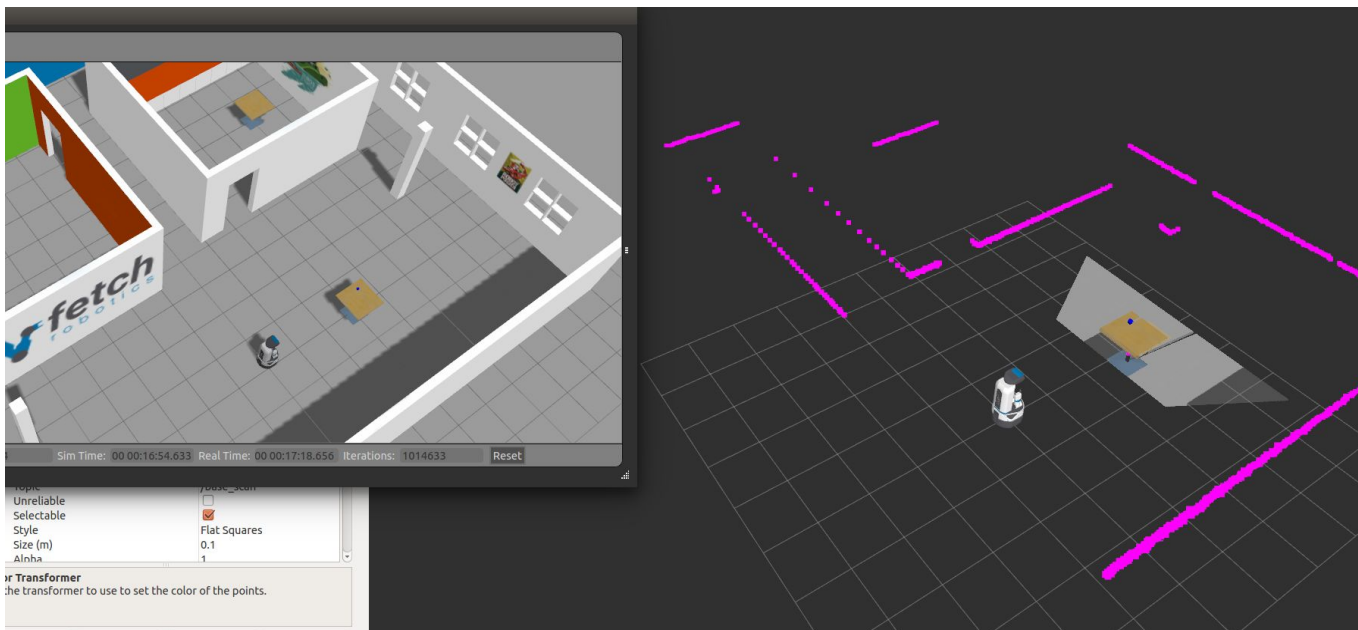
# Fetch Gazebo

- Simulation of a Fetch robot in a world with obstacles



# Rviz

- Visualize ROS messages
  - E.g. laser scan, point clouds



# catkin

- catkin is a build system for ROS
  - Similar to cmake
- catkin\_workspace
  - Run “catkin\_make” to build all ros packages in the src/ folder

```
workspace_folder/      -- WORKSPACE
  src/                 -- SOURCE SPACE
    CMakeLists.txt     -- The 'toplevel' CMake file
    package_1/
      CMakeLists.txt
      package.xml
      ...
    package_n/
      CATKIN_IGNORE    -- Optional empty file to exclude package_n from being processed
      CMakeLists.txt
      package.xml
      ...
```

# Create a catkin\_workspace

```
$ source /opt/ros/indigo/setup.bash
```

```
$ mkdir -p ~/catkin_ws/src
```

```
$ cd ~/catkin_ws
```

```
$ catkin_make
```

# Create a ros package

```
$ cd ~/catkin_ws/src
```

```
$ catkin_create_pkg ros_tutorial
```

```
ros_tutorial/  
  CMakeLists.txt  
  package.xml
```

```
$ cd ros_tutorial
```

```
$ mkdir src
```

```
$ cd src
```



# ROS node

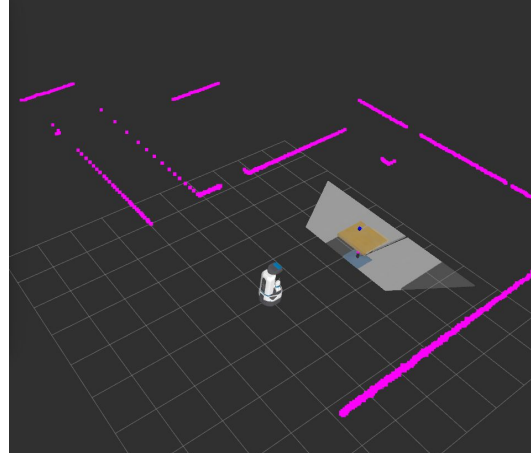
- A ROS node is not much more than an executable file within ROS package
  - ROS nodes use a ROS client library to communicate with other nodes
    - roscpp (c++ client library for ROS)
    - rospy (Python client library for ROS)
  - `$ rosnodetool list`
- ROS nodes can
  - Publish a ROS Topic
  - Subscribe to a ROS Topic
  - Provide a ROS Service
  - Use a ROS Service

# ROS Topic

- Remotely log into the machine
  - `ssh UNIQUENAME@AP_IP`
  - Password: autorob
- Checkout ros topics
  - `rostopic list`
  - `rostopic echo cmd_vel`
  - `rostopic hz base_scan`
  - `rostopic type base_scan`
  - `rosmmsg show [base_scan_type]`

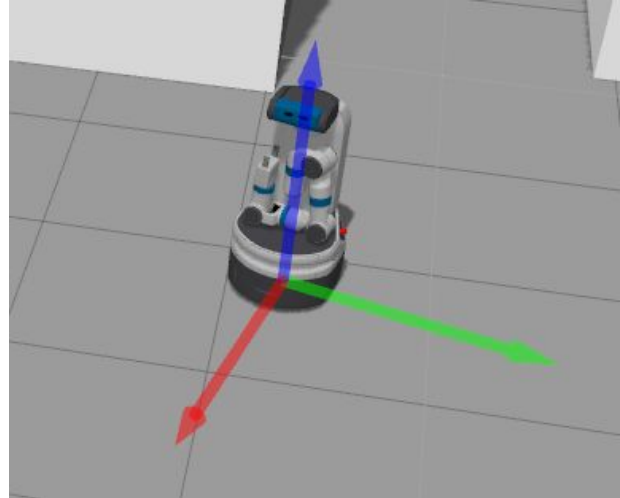
# Practice - Subscribe to base\_scan

- Write a ROS subscriber
  - Topic name
  - Subscriber callback function
  - Message type
- Build your subscriber node
  - `find_package(catkin REQUIRED COMPONENTS roscpp [MESSAGE PACKAGE])`
  - `add_executable([ROS NODE NAME] [SRC_FILE])`
  - `target_link_libraries([ROS NODE NAME] ${catkin_LIBRARIES})`
  - `$ catkin_make`
- Run your subscriber node
  - `$ source ~/catkin_ws/devel/setup.bash`
  - `$ rosrn ros_tutorial [ROS NODE NAME]`



# Practice - Publisher to cmd\_vel

- Write a ROS publisher
  - Topic name
  - Message type
  - Publish rate
- Build your publisher node
  - `find_package(catkin REQUIRED COMPONENTS roscpp [MESSAGE PACKAGE])`
  - `add_executable([ROS NODE NAME] [SRC_FILE])`
  - `target_link_libraries([ROS NODE NAME] ${catkin_LIBRARIES})`
  - `$ catkin_make`
- Run your publisher node
  - `$ source ~/catkin_ws/devel/setup.bash`
  - `$ rosrn ros_tutorial [ROS NODE NAME]`



# Walk robot around

- Move the robot around without colliding with objects
  - Combine publisher and subscriber
  - Turn the robot when base\_scan shows robot is close to objects
  - Otherwise move the robot forward

