ROBOTICS WORKSHOP

4th November 2017

AIMS

- Basic understanding of how ROS works
- Create a simple-publisher subscriber communication
- Run a simple program and be able to modify it

SUMMARY

ONE-LINE INSTALLATION

- sudo curl https://raw.githubusercontent.com/oroca/oroca-ros-pkg/master/ros_install.sh | bash -s catkin_ws kinetic

kinetic/ Installation/ Ubuntu

Ubuntu install of ROS Kinetic

We are building Debian packages for several Ubuntu platforms, listed below. These packages are more efficient than source-based builds and are our preferred installation method for Ubuntu. Note that there are also packages available from Ubuntu upstream. Please see UpstreamPackages to understand the difference.

Ubuntu packages are built for the following distros and architectures.

Distro	amd64	i386	armhf
Wily	X	X	
Xenial	X	X	X

If you need to install from source (**not recommended**), please see source (download-and-compile) installation instructions.

If you rely on these packages, please support OSRF.

These packages are built and hosted on infrastructure maintained and paid for by the Open Source Robotics Foundation, a 501(c)(3) non-profit organization. If OSRF were to receive one penny for each downloaded package for just two months, we could cover our annual costs to manage, update, and host all of our online services. Please consider donating to OSRF today.

Contents

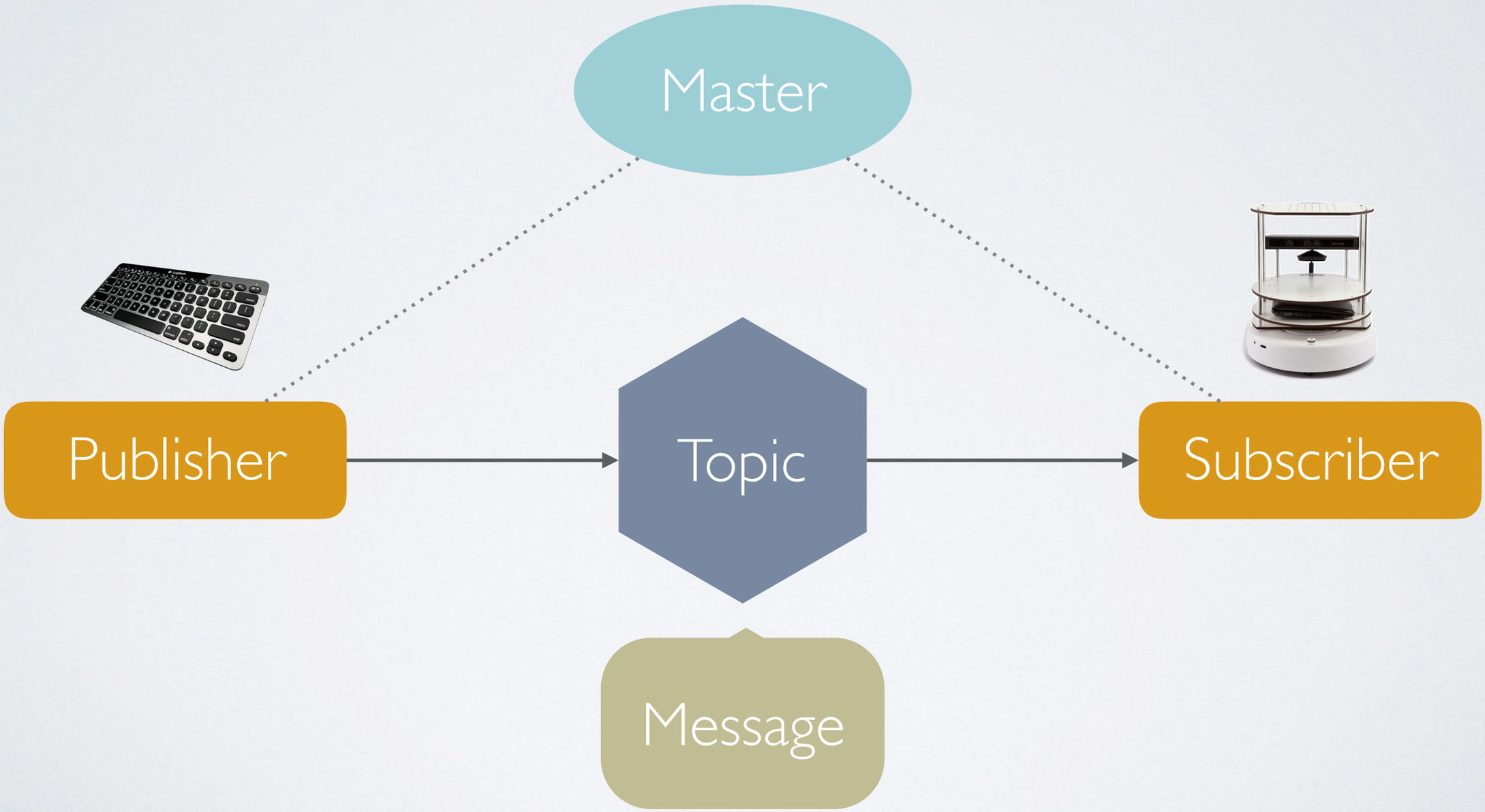
IN THE MEANTIME


Boston Dynamics


ASIMO


amazon
fulfillment

ROS



INSTALL TURTLEBOT

- sudo apt-get install ros-kinetic-turtlebot*
- roslaunch turtlebot_gazebo turtlebot_world.launch
- (another terminal) roslaunch turtlebot_teleop keyboard_teleop.launch —screen

Gazebo

World | Insert

- Scene
- Physics
- Models
  - bookshelf
  - jersey_barrier
  - ground_plane_0
  - unit_cylinder_1
  - Dumpster
  - cube_20k
  - mobile_base
- Lights

Property	Value
name	ground_plane_0
is_static	True
pose	
link	ground_plane_0...

Steps: 1 Real Time Factor: 1.01 Sim Time: 00 00:01:22.540 Real Time: 00 00:01:23.714 Iterations: 8254 Reset

GATHER INFO

- sudo apt-get install ros-kinetic-multimaster-fkie

- “node_manager”

Node Manager

22:23

ROS Network [id: 0]

localhost (localhost)

Start

/gazebo

Launch files

/gazebo

Settings

Help

/gazebo

URI: http://localhost:44833/
PID: 18552
ORG.MASTERURI: http://localhost:11311/

Subscribed Topics: [6]
_1_/ /clock *echo*
_0_/ ~set_link_state *echo*
_0_/ ~set_model_state *echo*
_0_/ /mobile_base/commands/motor_power *echo*
_0_/ /mobile_base/commands/reset_odometry *echo*
_1_/ /mobile_base/commands/velocity *echo*

Published Topics: [28]
_0_/ /camera/depth/camera_info *echo*
_0_/ /camera/depth/image_raw *echo*
_0_/ /camera/depth/points *echo*
_0_/ /camera/parameter_descriptions *echo*
_0_/ /camera/parameter_updates *echo*
_0_/ /camera/rgb/camera_info *echo*
_0_/ /camera/rgb/image_raw *echo*
_0_/ /camera/rgb/image_raw/compressed *echo*
_0_/ /camera/rgb/image_raw/compressed/
parameter_descriptions *echo*

localhost
updated: 22:23:27 (01 sec)

nodes filter

Name	Cfgs
localhost@localhost	
/bumper2pointcloud	
/cmd_vel_mux	
/depthimage_to_laserscan	
/gazebo	
/laserscan_nodelet_manager	
/mobile_base_nodelet_manager	
/robot_state_publisher	
/turtlebot_teleop_keyboard	
{SYSTEM}	

Nodes | Topics | Services | Parameter

Expert View

Capabilities View

Node Manager

22:24

ROS Network [id: 0]

localhost (localhost)

Start

/turtlebot_teleop_keyboard

/turtlebot_teleop_keyboard

URI: http://localhost:39321/
PID: 19571
ORG.MASTERURI: http://localhost:11311/

Subscribed Topics:
_1_/ /clock *echo*

Published Topics: [2]
_1_/ /cmd_vel_mux/input/teleop *echo*
_2_/ /rosout *echo*

Services: [2]
~get_loggers *call*
~set_logger_level *call*

Launch files | /turtlebot_teleop_keyboard | Settings | Help

localhost
updated: 22:24:28 (12 sec)

nodes filter

Name	Cfgs
localhost@localhost	
/bumper2pointcloud	
/cmd_vel_mux	
/depthimage_to_laserscan	
/gazebo	
/laserscan_nodelet_manager	
/mobile_base_nodelet_manager	
/robot_state_publisher	
/turtlebot_teleop_keyboard	
{SYSTEM}	

Nodes | Topics | Services | Parameter

Expert View | Capabilities View

Node Manager

22:27

ROS Network [id: 0]

localhost (localhost)

Start

/cmd_vel_mux/input/teleop

/cmd_vel_mux/input/**teleop**

Subscriber:
/mobile_base_nodelet_manager

Publisher:
/turtlebot_teleop_keyboard

Type: geometry_msgs/Twist

linear: geometry_msgs/Vector3
angular: geometry_msgs/Vector3

Launch files
/cmd_vel_mux/input/teleop
Settings
Help

localhost
updated: 22:27:00 (16 sec)

nodes filter

Name	Cfgs
localhost@localhost	
/bumper2pointcloud	
/cmd_vel_mux	
/depthimage_to_laserscan	
/gazebo	
/laserscan_nodelet_manager	
/mobile_base_nodelet_manager	
/robot_state_publisher	
/turtlebot_teleop_keyboard	
{SYSTEM}	

Nodes | Topics | Services | Parameter

Expert View
Capabilities View

geometry_msgs/Twist Message

File: geometry_msgs/Twist.msg

Raw Message Definition

```
# This expresses velocity in free space broken into its linear and angular parts.
Vector3  linear
Vector3  angular
```

Compact Message Definition

```
geometry_msgs/Vector3 linear
geometry_msgs/Vector3 angular
```

geometry_msgs/Vector3 Message

File: geometry_msgs/Vector3.msg

Raw Message Definition

```
# This represents a vector in free space.
# It is only meant to represent a direction. Therefore, it does not
# make sense to apply a translation to it (e.g., when applying a
# generic rigid transformation to a Vector3, tf2 will only apply the
# rotation). If you want your data to be translatable too, use the
# geometry_msgs/Point message instead.

float64 x
float64 y
float64 z
```

Compact Message Definition

```
float64 x
float64 y
float64 z
```

CREATE PUBLISHER

- http://wiki.ros.org/turtlesim/Tutorials/Moving%20in%20a%20Straight%20Line

- modify to make it go in circles

Toggle line numbers

```
#!/usr/bin/env python
import rospy
from geometry_msgs.msg import Twist

def move():
    # Starts a new node
    rospy.init_node('robot_cleaner', anonymous=True)
    velocity_publisher = rospy.Publisher('/turtle1/cmd_vel', Twist, queue_size=10)
    vel_msg = Twist()

    #Receiveing the user's input
    print("Let's move your robot")
    speed = input("Input your speed:")
    distance = input("Type your distance:")
    isForward = input("Foward?: ")#True or False

    #Checking if the movement is forward or backwards
    if(isForward):
        vel_msg.linear.x = abs(speed)
    else:
        vel_msg.linear.x = -abs(speed)
    #Since we are moving just in x-axis
    vel_msg.linear.y = 0
    vel_msg.linear.z = 0
    vel_msg.angular.x = 0
    vel_msg.angular.y = 0
    vel_msg.angular.z = 0

    while not rospy.is_shutdown():

        #Setting the current time for distance calculus
        t0 = rospy.Time.now().to_sec()
```

TURTLEBOT RACE