

академия
больших
данных



mail.ru
group



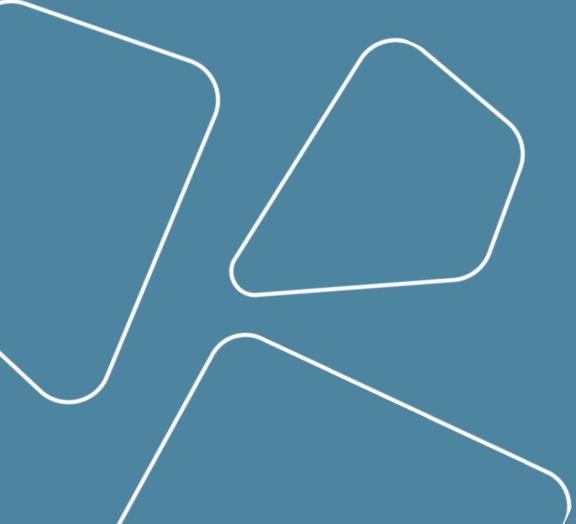
Introduction to Mobile Robotics course, Seminar 4

Names and Namespaces in ROS, time, debugging and visualization

Vladislav Goncharenko, Fall 2021

Materials by Oleg Shipitko

Outline

A graphic element in the bottom-left corner consisting of several white-outlined geometric shapes on a dark blue background. It includes a large irregular polygon, a triangle, and some curved lines.

7. Names and Namespaces
2. Time
3. Additional instruments
 - a. Tf
 - b. .bag files
 - c. rqt
4. Simulation
 - a. rviz
 - b. Gazebo

Names and Namespaces

**girafe
ai**

01

NAMES

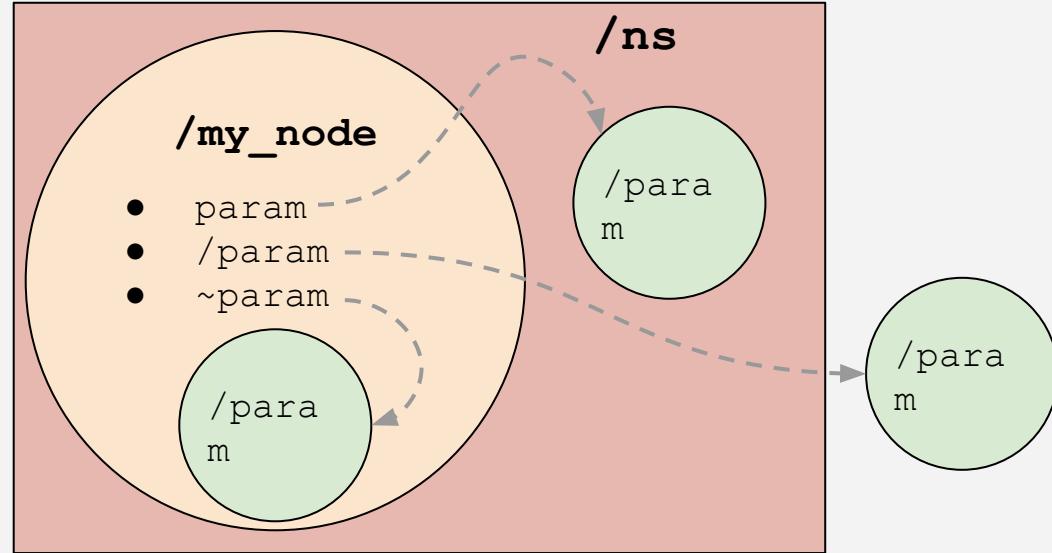
<http://wiki.ros.org/Names>

- ❑ In ROS you can name:
 - ❑ Node
 - ❑ Topic
 - ❑ Parameter
 - ❑ Service
 - ❑ ...
- ❑ Names and namespaces provide a mechanism of data encapsulation
- ❑ Naming rules:
 - ❑ First character: ([a-z|A-Z]), tilde (~) or forward slash (/)
 - ❑ Following symbols: ([0-9|a-z|A-Z]), underscore (_) or forward slash (/)

NAMES

<http://wiki.ros.org/Names>

- ❑ 4 name types:
 - ❑ relative/name
 - ❑ base
 - ❑ /global/name
 - ❑ ~private/name
- ❑ Rules of transform:



Node name	Relative name	Global name	Private name
<code>/my_node</code>	<code>param</code> -> <code>/param</code>	<code>/param</code> -> <code>/param</code>	<code>~param</code> -> <code>/my_node/param</code>
<code>/ns/my_node</code>	<code>param</code> -> <code>/ns/param</code>	<code>/param</code> -> <code>/param</code>	<code>~param</code> -> <code>/ns/my_node/param</code>
	<code>param_ns/param</code> -> <code>/ns/param_ns/param</code>	<code>/param_ns/param</code> -> <code>/param_ns/param</code>	<code>~param_ns/param</code> -> <code>/ns/my_node/param_ns/param</code>

Time

girafe
ai

02

TIME

<http://wiki.ros.org/rospy/Overview/Time>

- ❑ ROS uses UNIX-time as time stamps
 - ❑ **UNIX-time** – integer, which is incremented each second. Contains amount of seconds since 00:00:00 UTC January 1, 1970.
- ❑ Client libraries (rospy, roscpp, ...) provide an API for time-related functions:
 - ❑ Base classes `Time` & `Duration`, `Timer` with support of arithmetic operations
 - ❑ Functions to get system time
 - ❑ `rospy.sleep()` & `rospy.Rate.sleep()`

```
now = rospy.get_rostime() # equivalent now = rospy.Time.now()
rospy.loginfo("Current time %i %i", now.secs, now.nsecs)
```

```
two_hours = rospy.Duration(60*60) + rospy.Duration(60*60)
one_hour = rospy.Duration(2*60*60) - rospy.Duration(60*60)
tomorrow = rospy.Time.now() + rospy.Duration(24*60*60)
negative_one_day = rospy.Time.now() - tomorrow
```

```
# sleep for 10 seconds
rospy.sleep(10.)

# sleep for duration
d = rospy.Duration(10, 0)
rospy.sleep(d)
```

Additional instruments

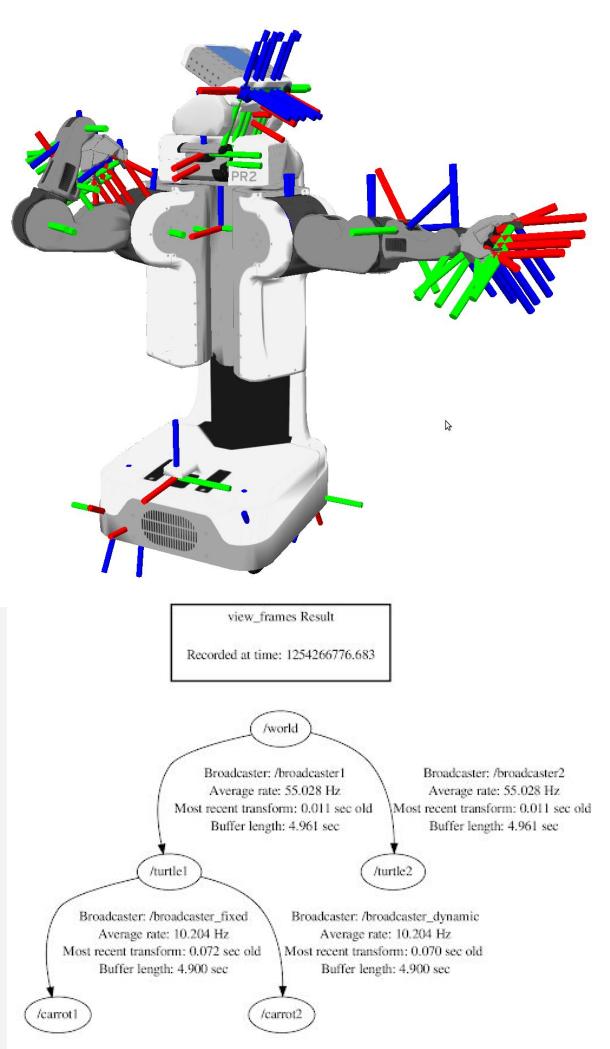
**girafe
ai**

03

TF

<http://wiki.ros.org/tf>
<http://wiki.ros.org/tf2>

- ❑ tf and tf2 packets contain:
 - ❑ Transforms between coordinate systems of robot in a form of a tree. It's also get buffered in time
 - ❑ Provide an interface to transform point, vectors, etc. between any coordinate system
 - ❑ Can be used to recover transform between coordinate systems of the past



ROS BAG

<http://wiki.ros.org/rosbag>

- .bag files contain **serialized** ROS-messages.
- .bag files can be played as topics which they contain
- Format of .bag files is **effective for recording** and **for playing**
because messages saved in the same format as in the ROS
network

HOW TO RECORD ROSBAG

```
rosbag record <topic-names>
```

-d, --duration

Maximum duration

```
$ rosbag record --duration=30 /chatter  
$ rosbag record --duration=5m /chatter  
$ rosbag record --duration=2h /chatter
```

--split

Split file if it exceeds maximum duration/size

```
$ rosbag record --split --size=1024 /chatter  
$ rosbag record --split --duration=30 /chatter  
$ rosbag record --split --duration=5m /chatter  
$ rosbag record --split --duration=2h /chatter
```

--max-splits=MAX_SPLITS

Same as split but when file has been splitted **MAX_SPLITS** times, it starts to remove old files

```
$ rosbag record --split --size 1024 --max-splits 3 /chatter  
$ rosbag record --split --duration 10m --max-splits 6 /chatter
```

-b SIZE, --buffsize=SIZE

Use internal buffer with SIZE MB (Default: 256, 0 = unlimited).

Creates message queue for object recorder. It gets filled before writing to the file. Decreasing buffer size can lead to messages loses

```
$ rosbag record -b 1024 /chatter
```

--chunksize=SIZE

Record data blocks of SIZE KB (Default: 768). Size of buffer of a bag file. Decreasing buffer size leads to more frequent writing on the disk.

```
$ rosbag record --chunksize=1024 /chatter
```

-l NUM, --limit=NUM

Record only NUM messages of each topic

```
$ rosbag record -l 1000 /chatter
```

--node=NODE

Record all the topics, NODE is subscribed on

```
$ rosbag record --node=/joy_teleop
```

-j, --bz2

Use BZ2 compression.

```
$ rosbag record -j /chatter
```

--lz4

Use LZ4 compression.

```
$ rosbag record --lz4 /chatter
```

WHAT'S INSIDE?

- ❑ `rosbag info <bag-files>`
- ❑ `rostopic list -b <bag-file>`
- ❑ `rostopic echo -b <bag-file>`

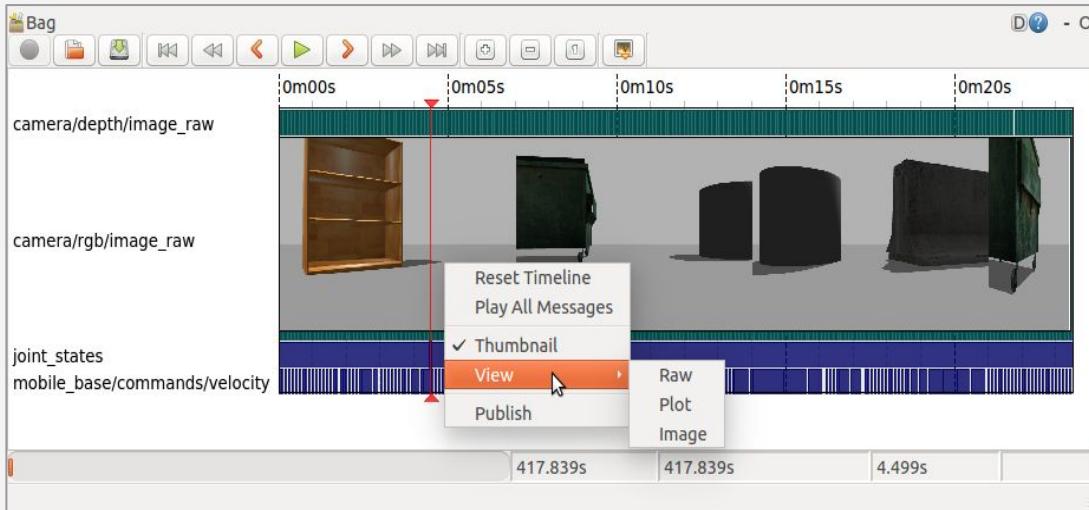
```
shipitko@devel-Latitude-5491: ~
shipitko@devel-Latitude-5491: ~ 115x67

~ rosbag info 2018-11-07-16-37-15_0.bag
path: 2018-11-07-16-37-15_0.bag
version: 2.0
duration: 2.06s (126s)
start: Nov 07 2018 16:37:15.38 (1541597835.38)
end: Nov 07 2018 16:39:21.63 (1541597961.63)
size: 272.6 MB
messages: 66604
compression: none [325/325 chunks]
types:
  a16_vision_msgs/RoadRecognitionResult [b04d56e923d6211527e5deac7cbe33ac]
  geometry_msgs/PoseStamped [d3812c3cbc69362b77dc0b19b345f87f5]
  motion_control_msgs/ControlMode [49e73fe9ca259cca296097e6eca9fd]
  motion_control_msgs/TopLevelControllerState2 [2753d177fe1e0a8df0bf696fc462d63]
  motion_control_msgs/VehicleDriverState [fdeb5fd8b57901f224d3939cbc2b51]
  motion_control_msgs/WaypointArray2 [a254e1443da343b241d358c8e42b660]
  rosgraph_msgs/Log [acfdd30c0bb0de30f120938c17c593fb]
  sensor_msgs/CameraInfo [c9a58c1b0b154e0e6da7578cb991d214]
  sensor_msgs/CompressedImage [8f7a12999da2c9d3352d540a977563f]
  sensor_msgs/Imu [6a62c6d0aa103f4ff7a132d6f95cec2]
  sensor_msgs/TimeReference [fded64a0265108ba86c3d38fb11c016]
  tf2_msgs/TFMessage [94810edd583a504d7fd3829e70d'eeec]
  vi_device_msgs/DriverTask [2d67f2c805e9965145802d04684e134]
  vi_device_msgs/OdometryExtended [7b1/d630a932660214780f1041298095]
  vi_device_msgs/RotationSensors2 [ffd143616a45ccb7e887fd9c143fc8cb]
  vi_device_msgs/SyncMultiRange [7fd69e282603679f336e115e5206396c]
  vi_device_msgs/VehicleTask2v3 [7d89c1149e46ee01b1bfe956a97d13]
  vi_nmea_msgs/Sentence [9f221efc5f4b3ba7ce4fa102b32308b]
  vi_nmea_navsat_driver/NavSatFixExtended [850a0ad46c4e59440c270781e931c7ef]
  walls_detection/WallsRecognitionResult [b1745e055246a775d31769d017453aa]
topics:
  /control/control_mode 1261 msgs : motion_control_msgs/ControlMode
  /control/top_level_control/state 1261 msgs : motion_control_msgs/TopLevelControllerState2
  /control/trajectory waypoints 1261 msgs : motion_control_msgs/WaypointArray2
  /depth_registered/compressedDepth 336 msgs : sensor_msgs/CompressedImage
  /driver/task 7355 msgs : vi_device_msgs/DriverTask
  /gnss/fix 126 msgs : vi_nmea_navsat_driver/NavSatFixExtended
  /gnss/mmea_sentence 792 msgs : vi_nmea_msgs/Sentence
  /gnss/time_reference 126 msgs : sensor_msgs/TimeReference
  /imu/xsens/imu 5045 msgs : sensor_msgs/Imu
  /left/camera_info 337 msgs : sensor_msgs/CameraInfo
  /left/image_rect_color/compressed 343 msgs : sensor_msgs/CompressedImage
  /map/pose 1261 msgs : geometry_msgs/PoseStamped
  /odometry/extended 12870 msgs : vi_device_msgs/OdometryExtended
  /odometry/rear_wheels2 7824 msgs : vi_device_msgs/RotationSensors2
  /pc/detector/walls 252 msgs : walls_detection/WallsRecognitionResult
  /right/camera_info 336 msgs : sensor_msgs/CameraInfo
  /right/image_rect_color/compressed 339 msgs : sensor_msgs/CompressedImage
  /rosout 59 msgs : rosgraph_msgs/Log
  (17 connections)
    /rosout_agg 29 msgs : rosgraph_msgs/Log
    /sonars/front/sync 1232 msgs : vi_device_msgs/SyncMultiRange
    /tf 18254 msgs : tf2_msgs/TFMessage
  (4 connections)
    /vehicle/state 2523 msgs : motion_control_msgs/VehicleDriverState
    /vehicle/task 2523 msgs : vi_device_msgs/VehicleTask2v3
    /vision/front/right/camera_info 1261 msgs : sensor_msgs/CameraInfo
    /vision/front/right/image/compressed 1261 msgs : sensor_msgs/CompressedImage
    /vision/front/right/road_recognition 337 msgs : a16_vision_msgs/RoadRecognitionResult
```

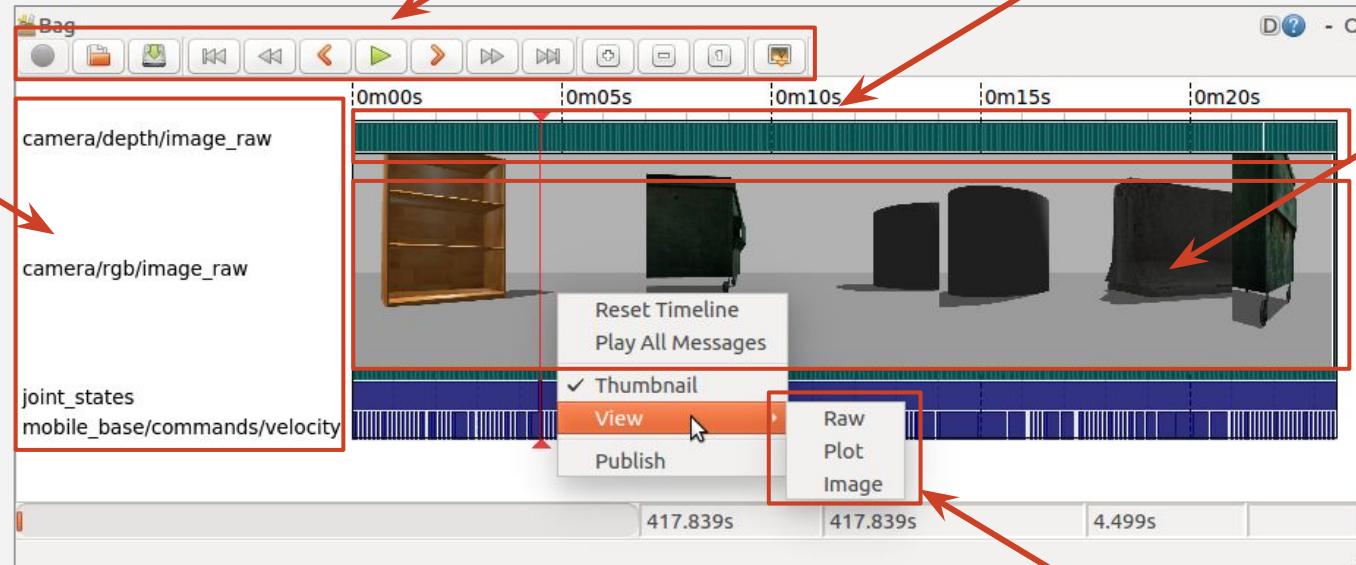
WHAT'S INSIDE?

rqt_bag packet

- ❑ Displays messages in topics
- ❑ Displays thumbnails изображения on the timeline
- ❑ Draws plots of numeric data
- ❑ Publishes/Records chosen topics
- ❑ Exports messages from selected time period to the new bag file



rqt_bag has an API, which you can use to develop your own plugins

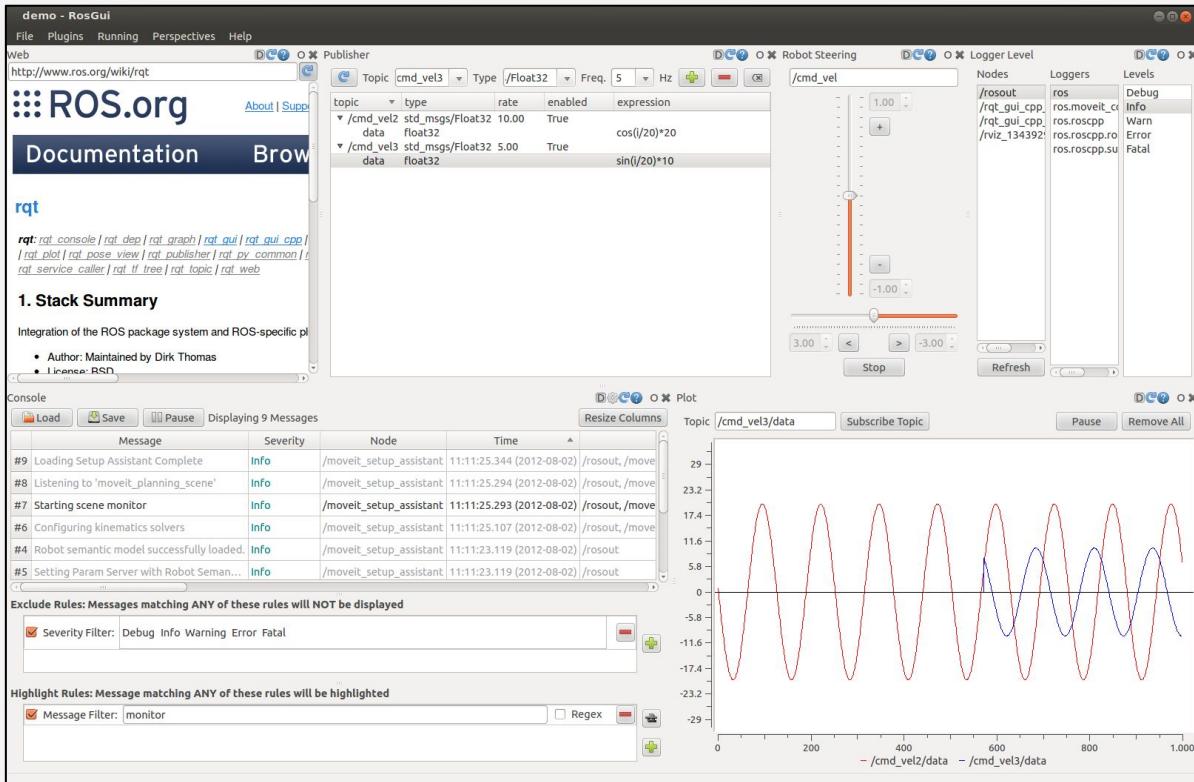


14

RQT

<http://wiki.ros.org/rqt>

- ❑ Framework used to create GUI ROS-apps on top of Qt
- ❑ Has a set of ready to use plugins
- ❑ Provides an API to develop your own plugins



Simulation

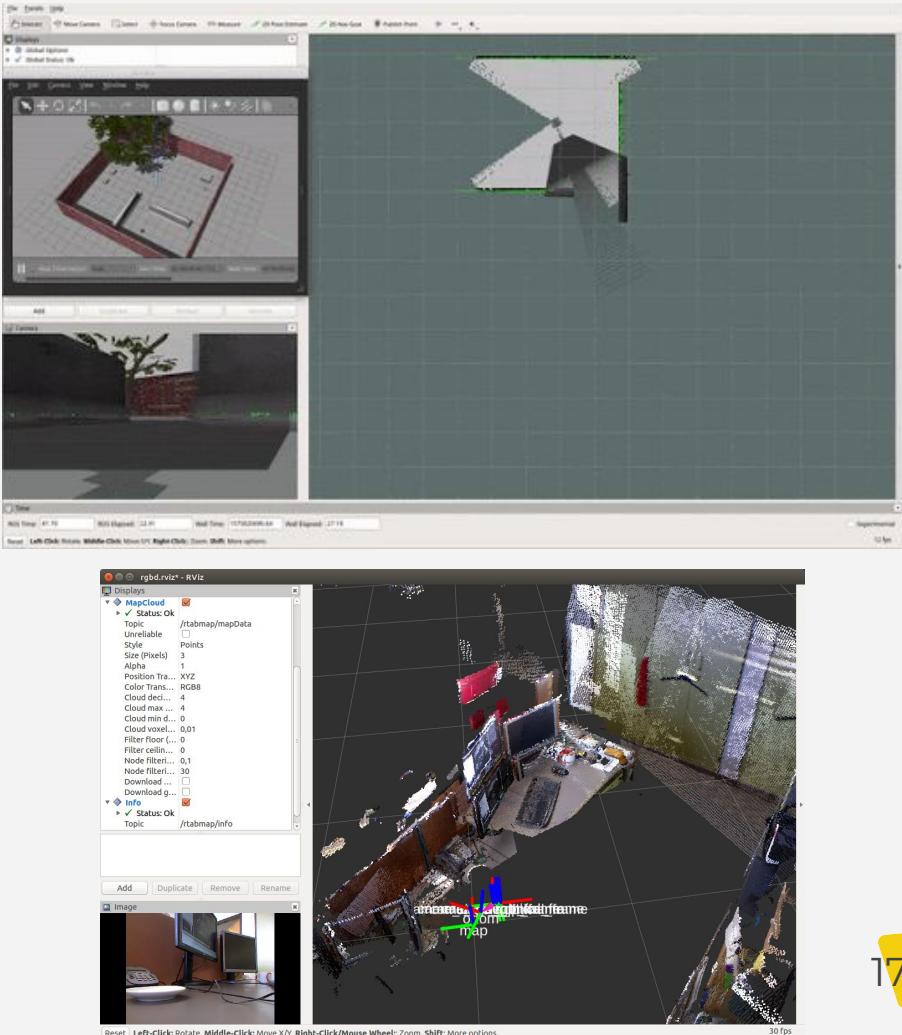
girafe
ai

04

RVIZ

<http://wiki.ros.org/rviz>

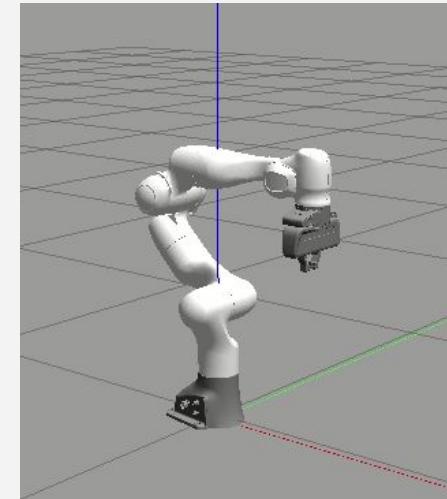
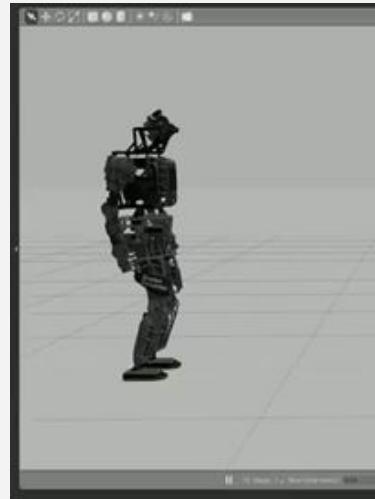
- ❑ ROS 2D/3D visualization tool
- ❑ Support visualization of commonly used types of data (occupation maps, laser scans, point clouds, coordinate systems, trajectories, etc.), as well as drawing graphical primitives (cubes, cylinders, points, lines, etc.) and even CAD-models
- ❑ Functionality can be extended by external plugins



GAZEBO

http://wiki.ros.org/gazebo_ros_pkgs

- ❑ **Gazebo** – open source 3D simulator of solid bodies
- ❑ Often used in conjunction with ROS to model robots and in robotics challenges
- ❑ Gazebo is able to:
 - ❑ Use different physical engines: ODE, Bullet, etc.
 - ❑ Render realistic scenes with use of different light sources, shadows. Textures, etc.
 - ❑ Model sensors, including their measurement noises: LIDARs, cameras, depth cameras



TURTLEBOT

<http://wiki.ros.org/Robots/TurtleBot>

<http://emanual.robotis.com/docs/en/platform/turtlebot3/overview/#turtlebot>

Original TurtleBot

(Discontinued)



TurtleBot 2 Family



TurtleBot 2



TurtleBot 2i



TurtleBot 2e

TurtleBot 3 Family

Burger



Waffle



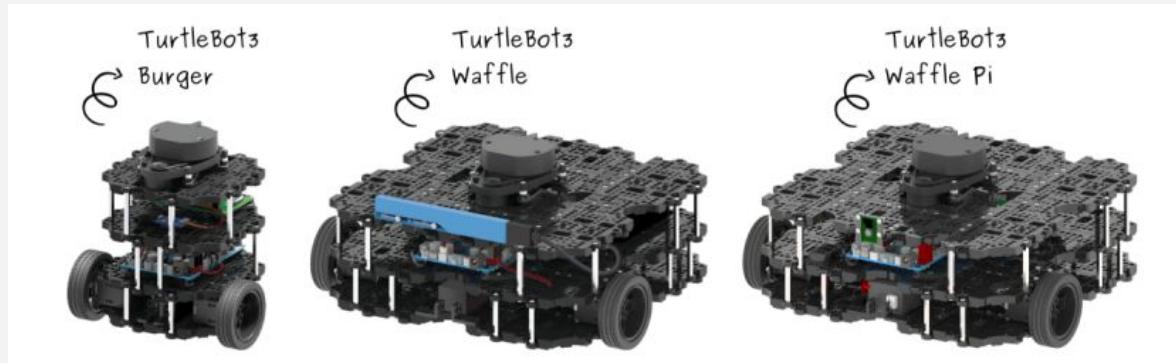
Waffle Pi



TURTLEBOT SIMULATION. PREPARATION

<http://emanual.robotis.com/docs/en/platform/turtlebot3/overview/#turtlebot>

- ❑ sudo apt update
- ❑ sudo apt install ros-melodic-turtlebot3
- ❑ cd /root/my_ros_ws/src
- ❑ git clone https://github.com/ROBOTIS-GIT/turtlebot3_simulations.git
- ❑ source /opt/ros/melodic/setup.zsh
- ❑ cd /root/my_ros_ws && catkin_make
- ❑ source ./devel/setup.zsh



TURTLEBOT SIMULATION. LAUNCH

<http://emanual.robotis.com/docs/en/platform/turtlebot3/overview/#turtlebot>

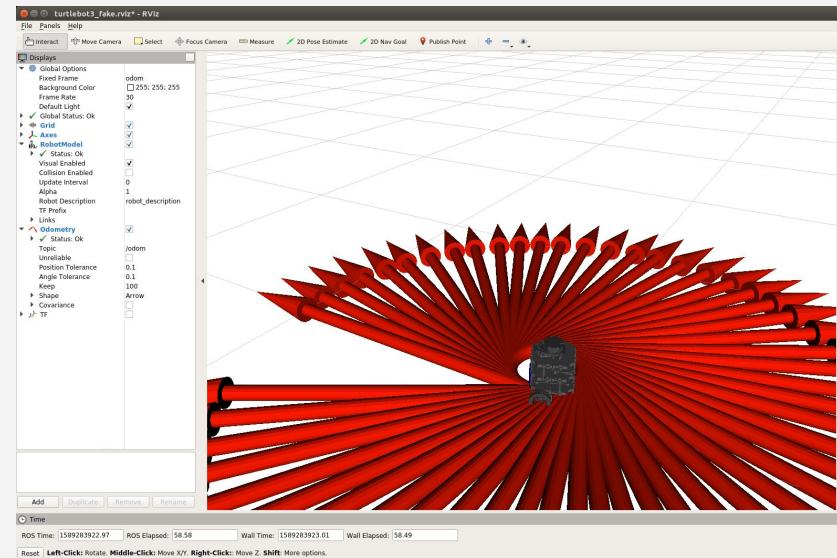
- ❑ `sudo apt update`

- ❑ First terminal:

- ❑ `export TURTLEBOT3_MODEL=${TB3_MODEL}`
 - ❑ `${TB3_MODEL}: burger, waffle, waffle_pi`
- ❑ `roslaunch turtlebot3_fake`
`turtlebot3_fake.launch`

- ❑ Second terminal:

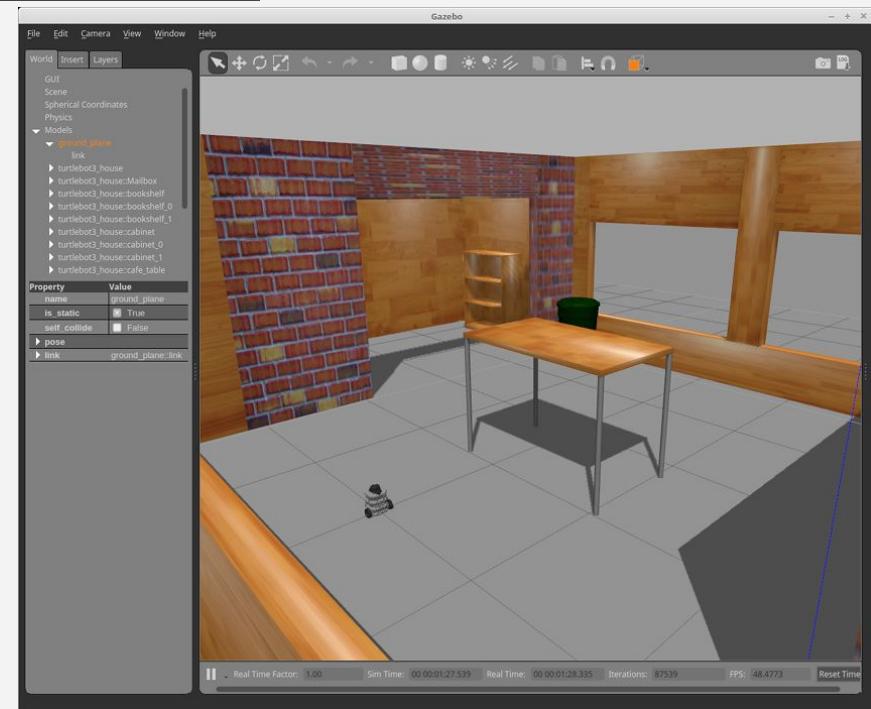
- ❑ `export TURTLEBOT3_MODEL=${TB3_MODEL}`
 - ❑ `${TB3_MODEL}: burger, waffle, waffle_pi`
- ❑ `roslaunch turtlebot3_teleop`
`turtlebot3_teleop_key.launch`



TURTLEBOT SIMULATION. COLLISION AVOIDANCE

<http://emanual.robotis.com/docs/en/platform/turtlebot3/overview/#turtlebot>

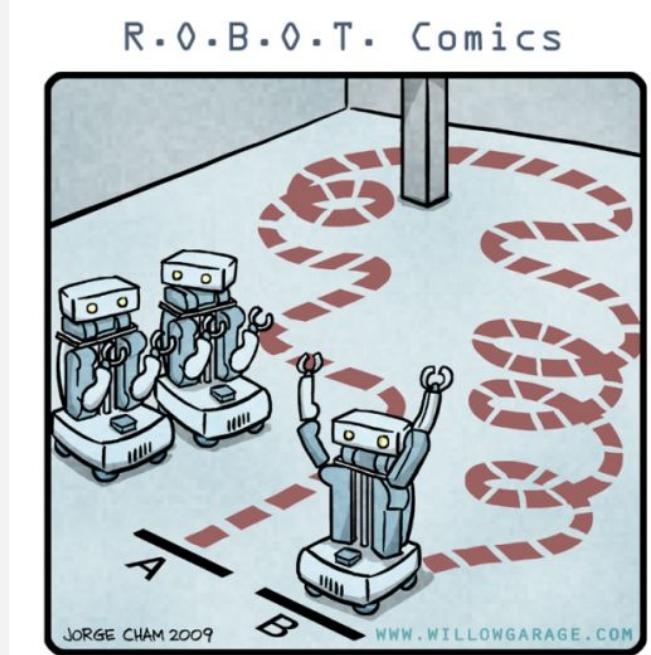
- ❑ В первом терминале:
 - ❑ `export TURTLEBOT3_MODEL=waffle_pi`
 - ❑ `roslaunch turtlebot3_gazebo turtlebot3_house.launch`
- ❑ В другом терминале:
 - ❑ `export TURTLEBOT3_MODEL=waffle_pi`
 - ❑ `roslaunch turtlebot3_gazebo turtlebot3_simulation.launch`
- ❑ В третьем:
 - ❑ `export TURTLEBOT3_MODEL=waffle_pi`
 - ❑ `roslaunch turtlebot3_gazebo turtlebot3_gazebo_rviz.launch`



ROS NAVIGATION STACK

<http://wiki.ros.org/navigation>

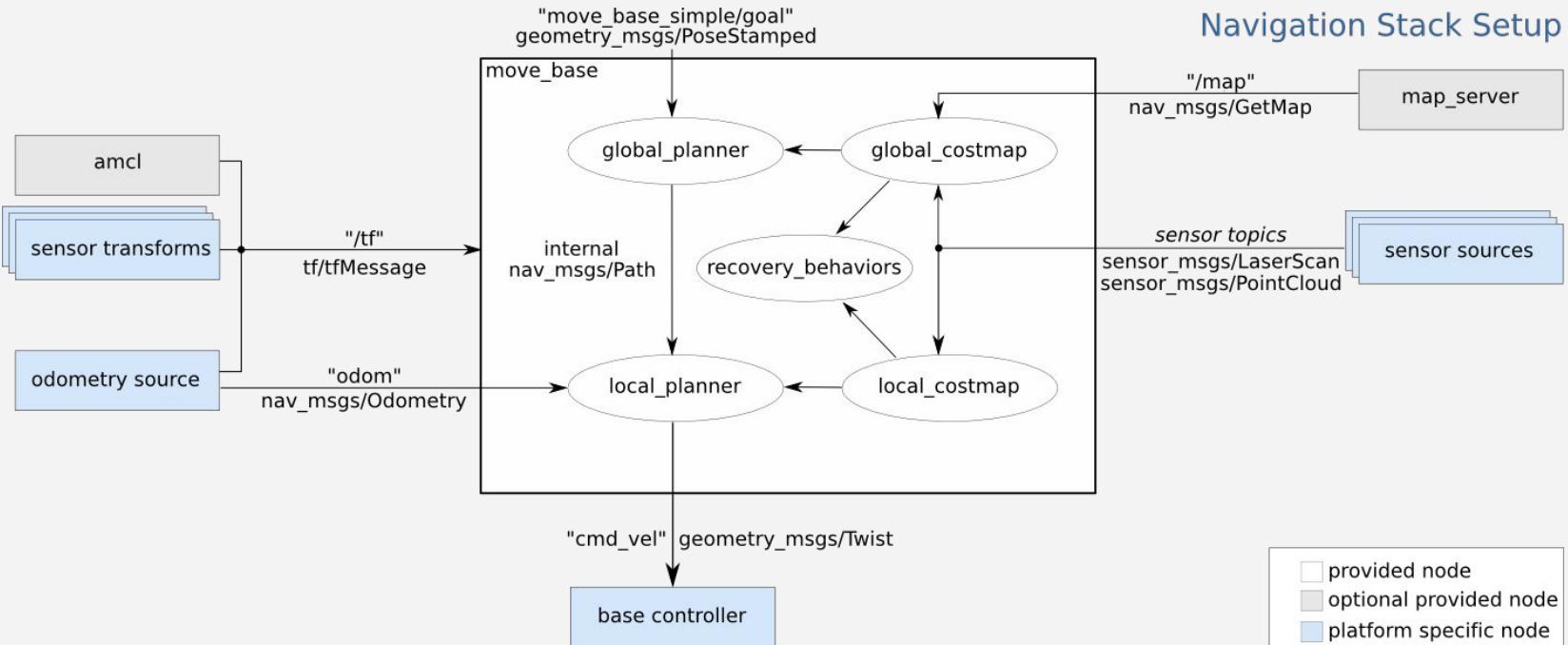
- ❑ 2D navigation stack gets **odometry, sensors data and goal position** as an input and calculates **speed control** (linear and angular), which will let robot get to the goal
- ❑ Limitations:
 - ❑ Intended to be used with **differential steering** robots or **holonomic** robots
 - ❑ Robot has to have planar LIDAR (or another sensor, which generate 2D scans) for map generation and localization
 - ❑ Fits to square or circular base robots. For robots with other forms of the base planning can be suboptimal



"HIS PATH-PLANNING MAY BE
SUB-OPTIMAL, BUT IT'S GOT FLAIR."

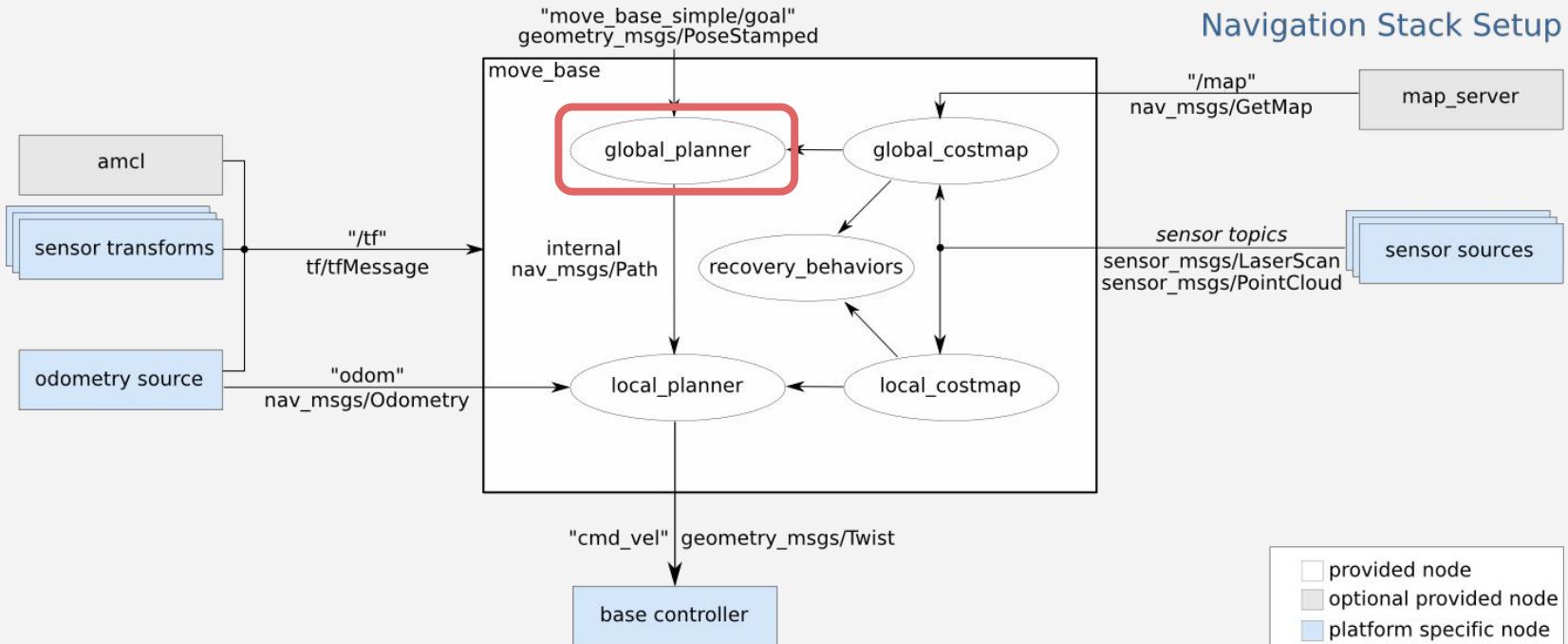
ROS NAVIGATION STACK

<http://wiki.ros.org/navigation>



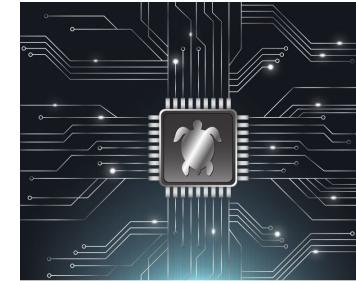
ROS NAVIGATION STACK

<http://wiki.ros.org/navigation>



ADDITIONAL RESOURCES

1. Book: [ROS Robot Programming](#).
YoonSeok Pyo, HanCheol Cho,
RyuWoon Jung, TaeHoon Lim
2. [ROS Official Tutorials](#)
3. [Clearpath Robotics ROS Tutorial](#)
4. [The history of ROS creation](#)



ROS
Robot Programming

From the basic concept to practical programming and robot application.

A Handbook Written by TurtleBot3 Developers

YoonSeok Pyo | HanCheol Cho | RyuWoon Jung | TaeHoon Lim

Thanks for attention!

Questions? Additions? Welcome!

girafe
ai

