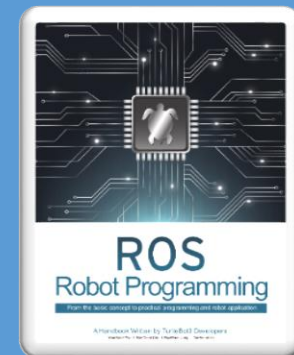


ROS Tools

ROBOTIS

KAIST



You Tube

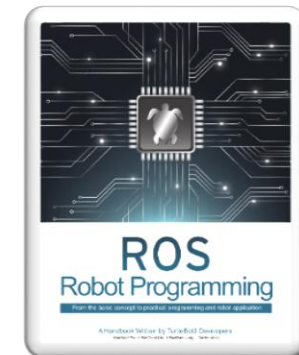
 **Subscribe**

Textbook
P. 128~147

Index

I. 3D Visualization Tool (Rviz)

II. ROS GUI Development Tool (rqt)



You Tube

Subscribe

Textbook
P. 128~147

Various Development Tools for ROS

- Provides various development tools needed for robot development
- Improving the efficiency of robot development

■ Command-Line Tools

- Robot access only with commands provided by ROS without GUI & Use almost all ROS features

■ RViz

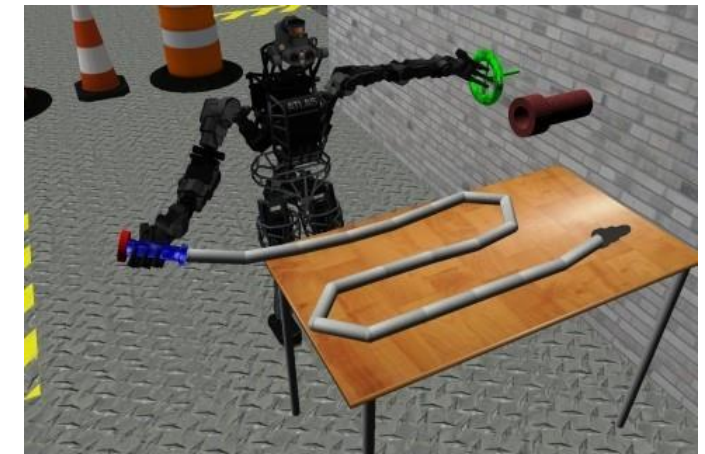
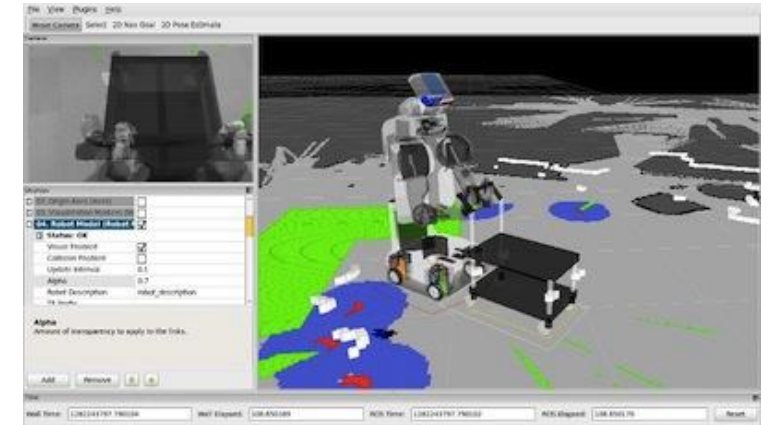
- Provides powerful 3D visualization tool
- Visualizes sensor data such as laser, camera, etc.
- Represents robot configuration and planned motion

■ RQT

- Provides Qt-based framework for developing graphic interface
- Displays nodes and connection information between them (rqt_graph)
- Floats encoder, voltage, or number that changes over time (rqt_plot)
- Records data in message form and play back (rqt_bag)

■ Gazebo

- 3D simulator which supports physics engine, robot, sensor, environmental model, etc.
- High compatibility with ROS



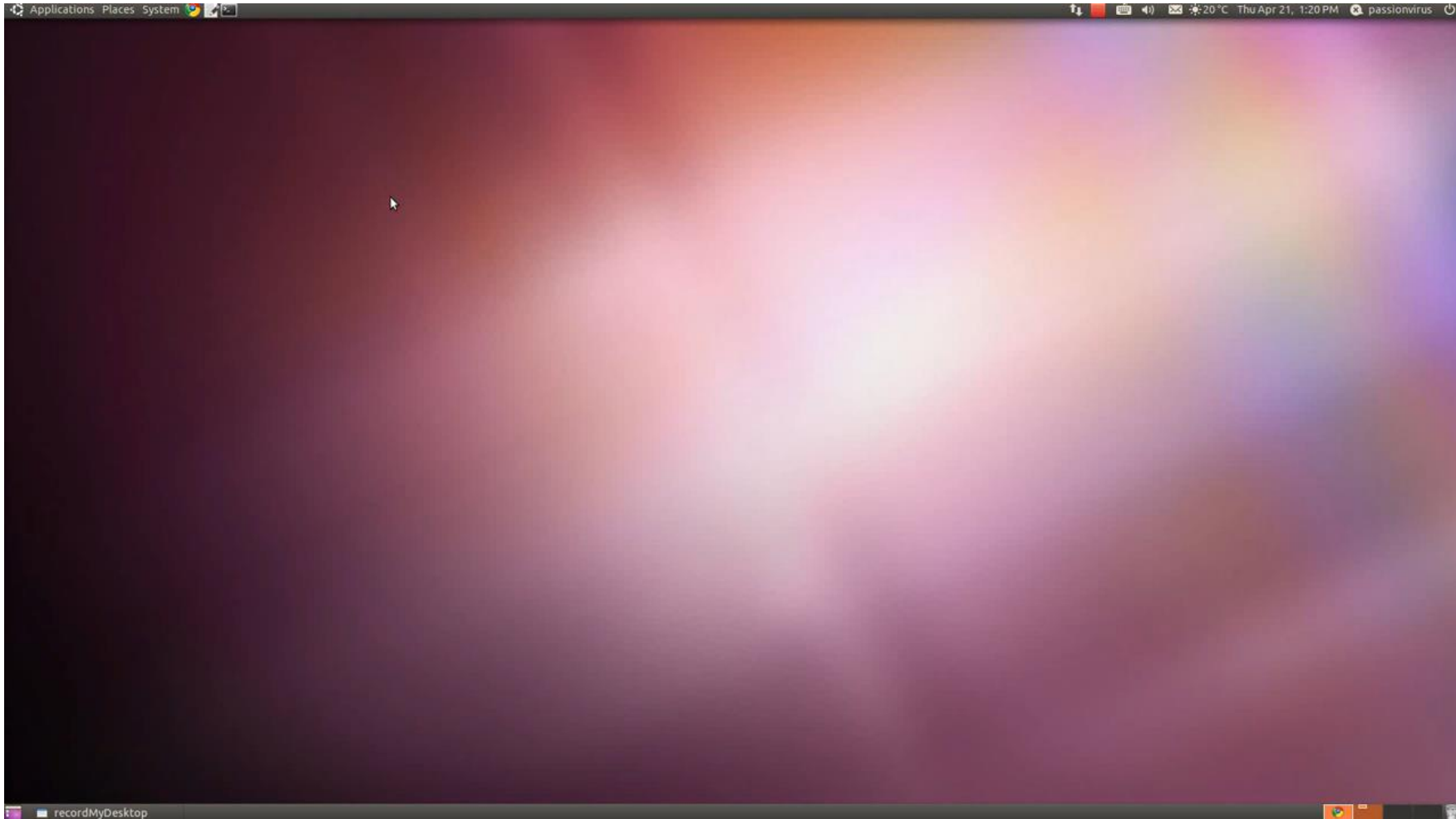
visualization Tool: RViz

RViz (ROS Visualization Tool)

- 3D visualization tool for ROS
 - Visualization of sensor data
 - Distance data of LASER distance sensor(LDS)
 - Point cloud data from depth camera such as 'RealSense', 'Kinect', 'Xtion', etc.
 - Image data of camera
 - Inertia data of IMU sensor ...
- Represents robot configuration and planned motion
 - URDF (Unified Robot Description Format)
- Navigation
- Manipulation
- Tele-operation

RViz Example #1

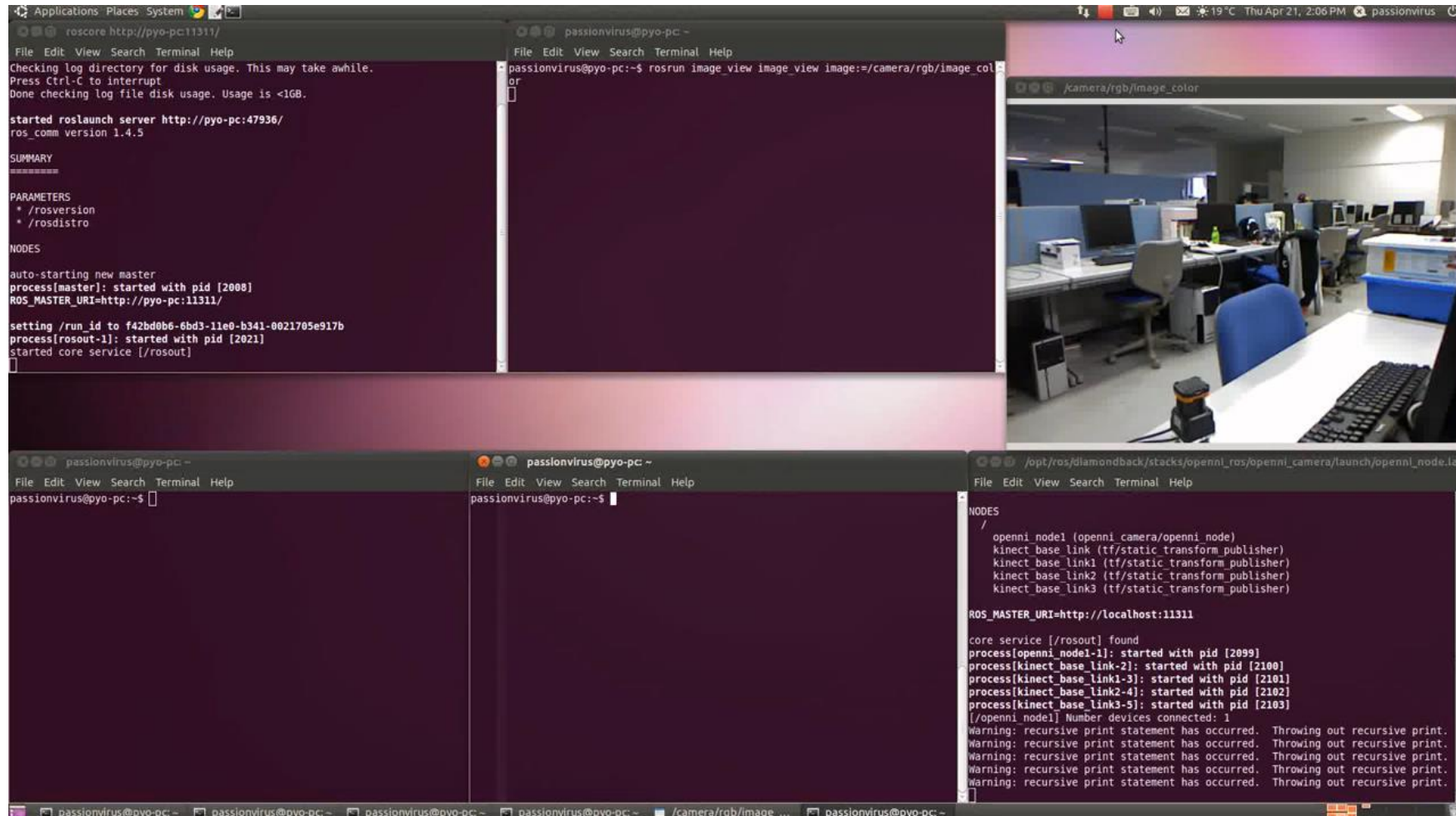
- Point cloud data of 'Kinect'



<https://youtu.be/OqOkpZBOpxY>

RViz Example #2

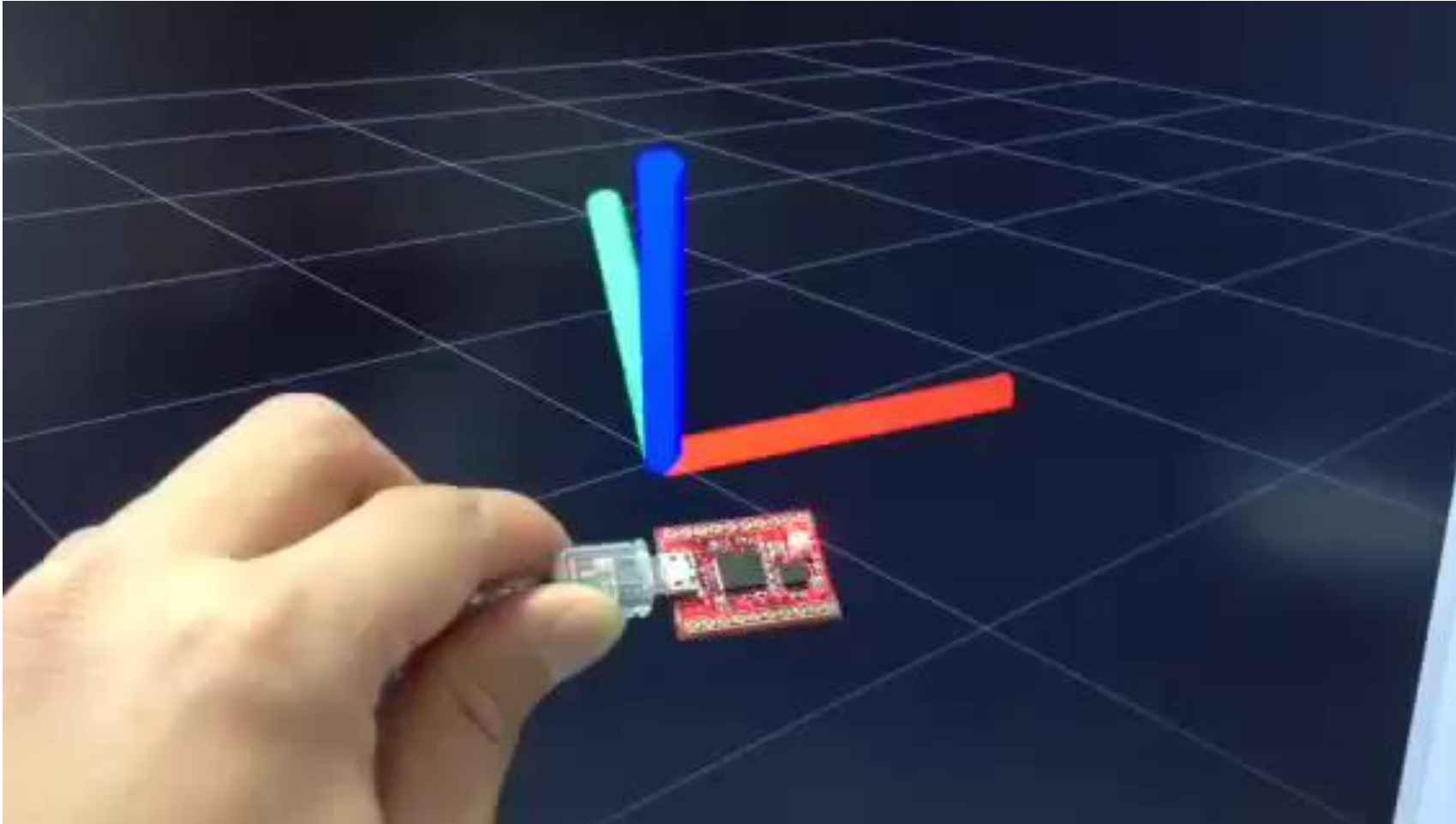
- Distance data of LASER distance sensor(LDS)



<https://youtu.be/qtoAJ1wzB6s>

RViz Example #3

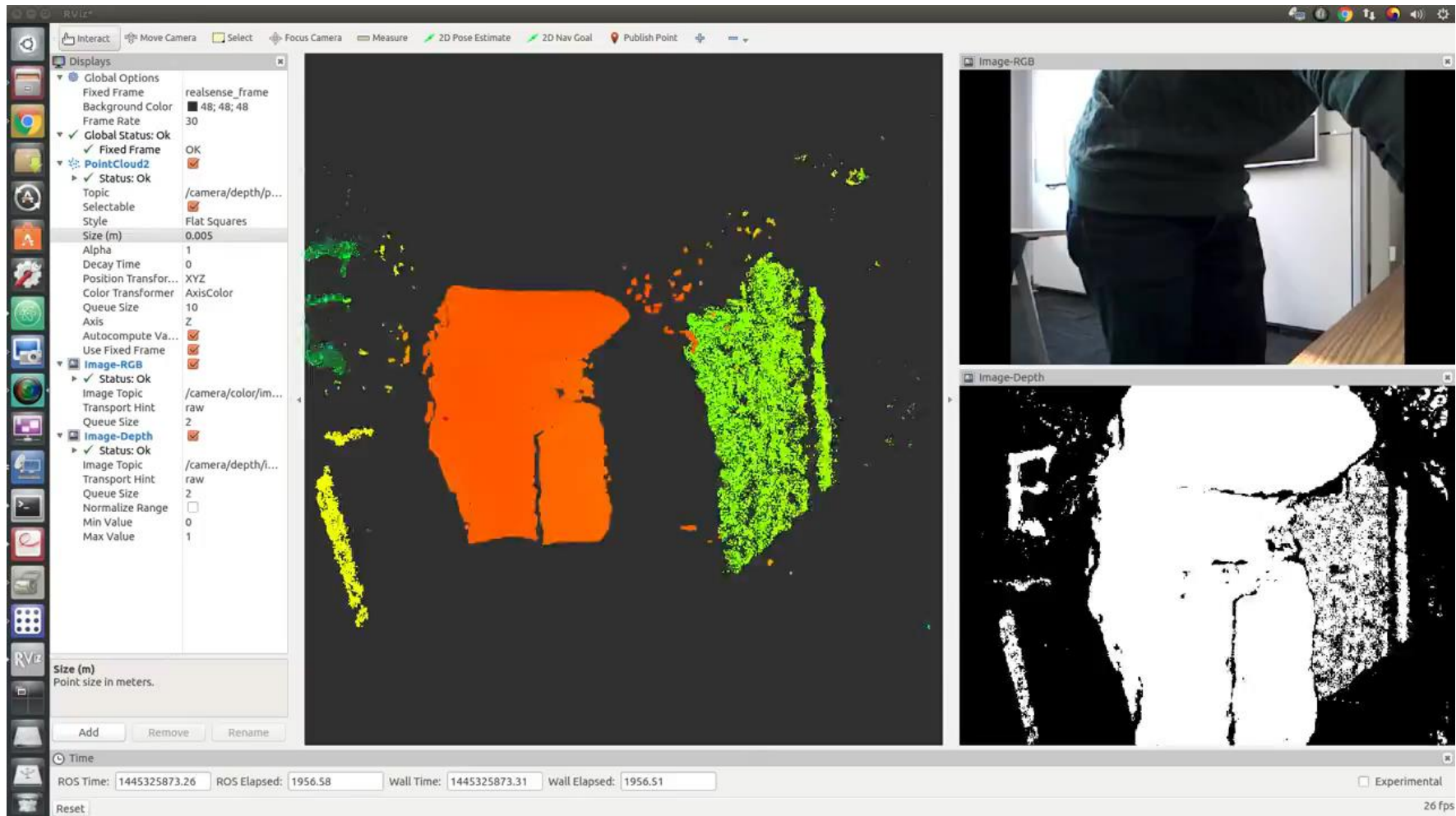
- Inertia data of IMU sensor



<https://youtu.be/j5v5fKppcQo>

RViz Example #4

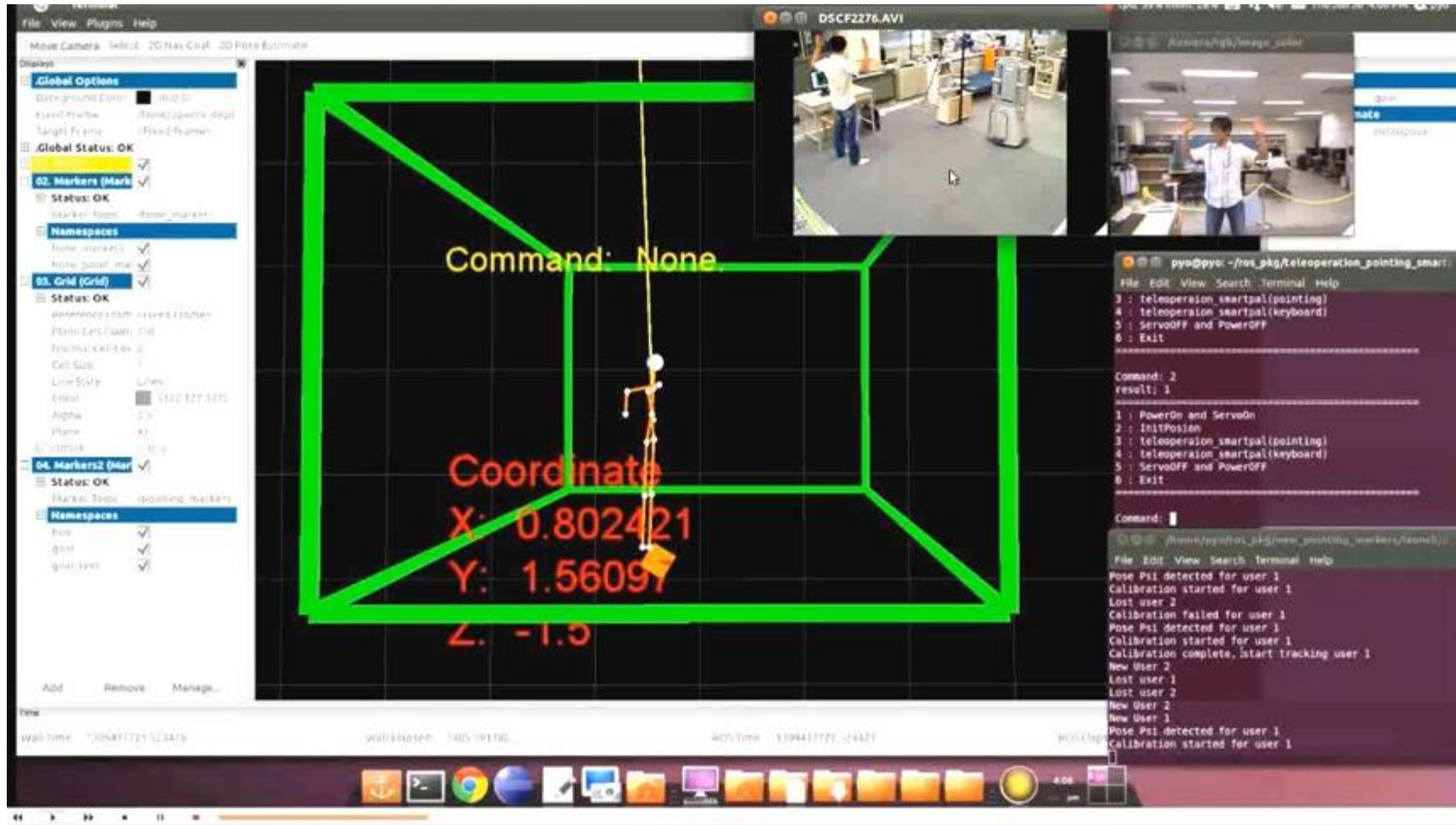
- Video of point cloud, color, depth of 'RealSense'



<https://youtu.be/Jf4kgPEzY4s>

RViz Example #5

- Show skeleton and direction of a person

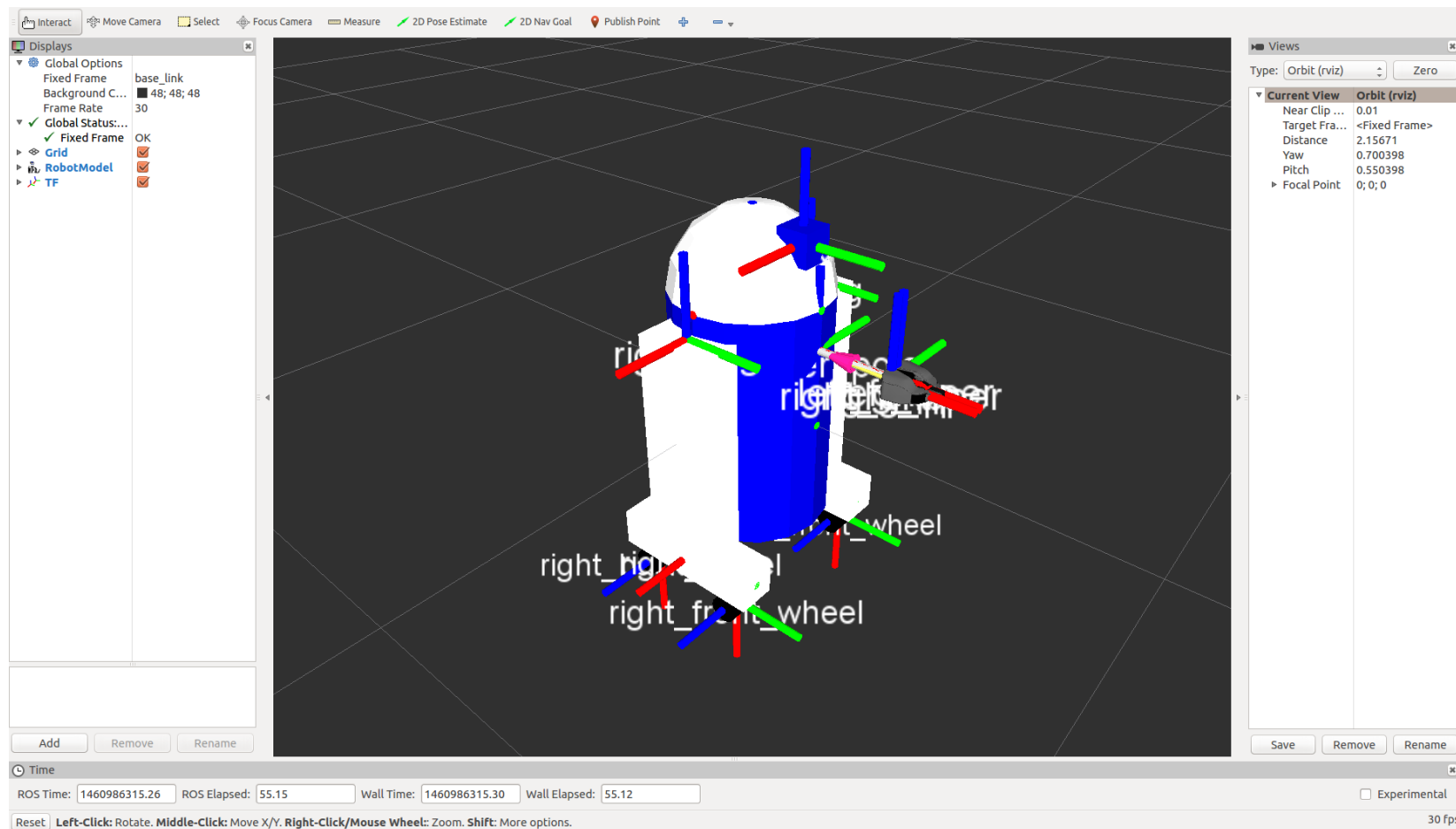


RViz Example #6

- R2-D2 robot model

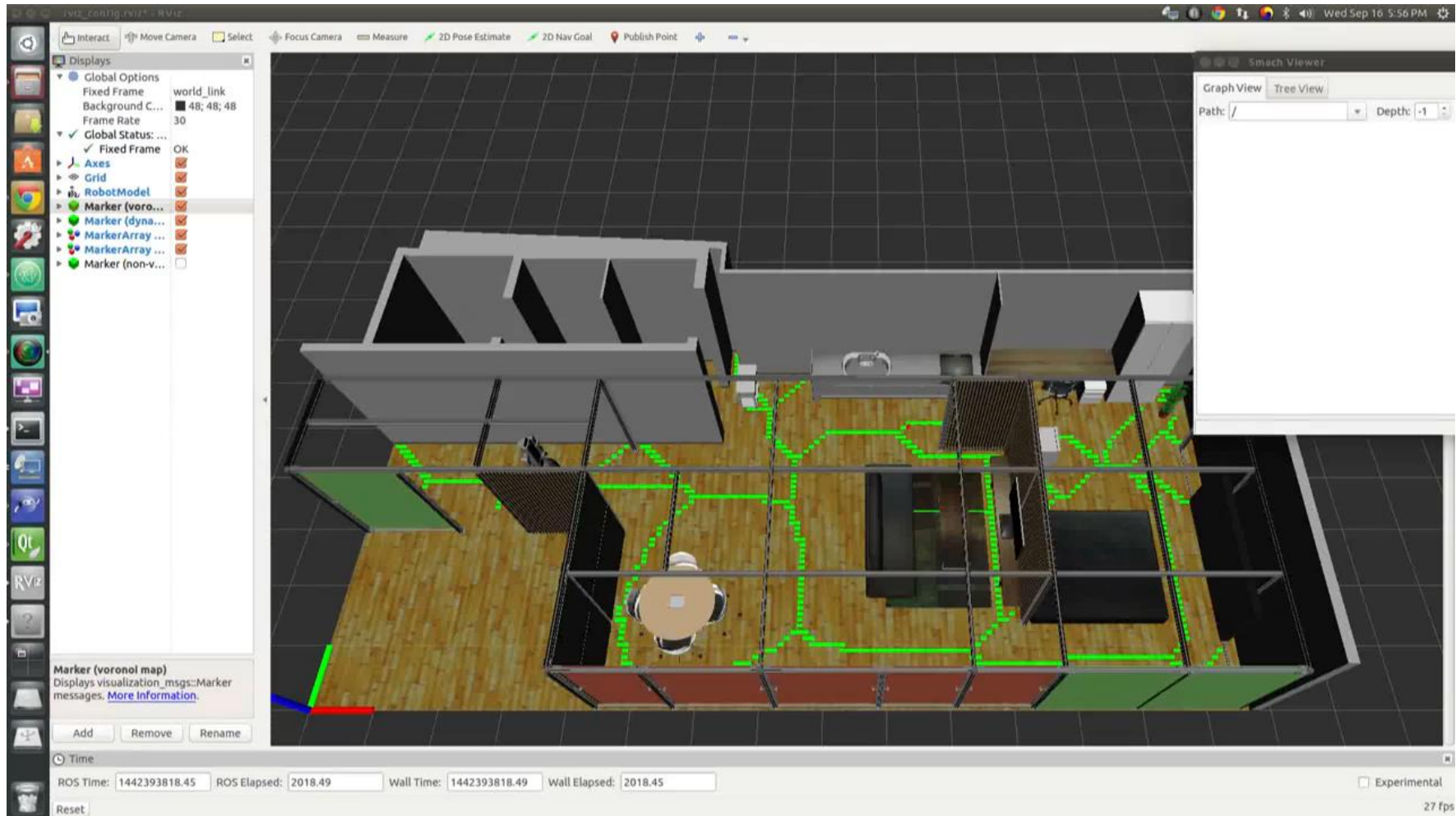
```
$ sudo apt-get install ros-kinetic-urdf-tutorial
```

```
$ roslaunch urdf_tutorial display.launch model:='$(find urdf_tutorial)'urdf/05-visual.urdf
```



RViz Example #7

- Environment model, robot, path



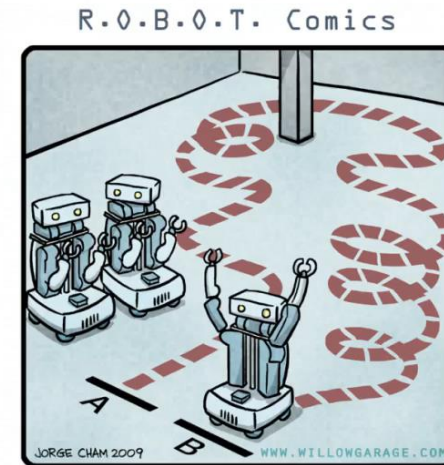
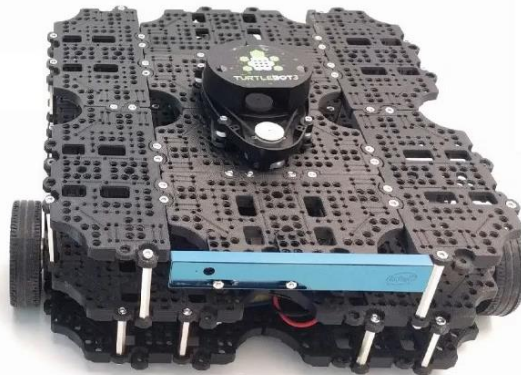
RViz Example #8

- Map display, navigation, and assign destination

TurtleBot3
BURGER ↻



TurtleBot3
WAFFLE ↻

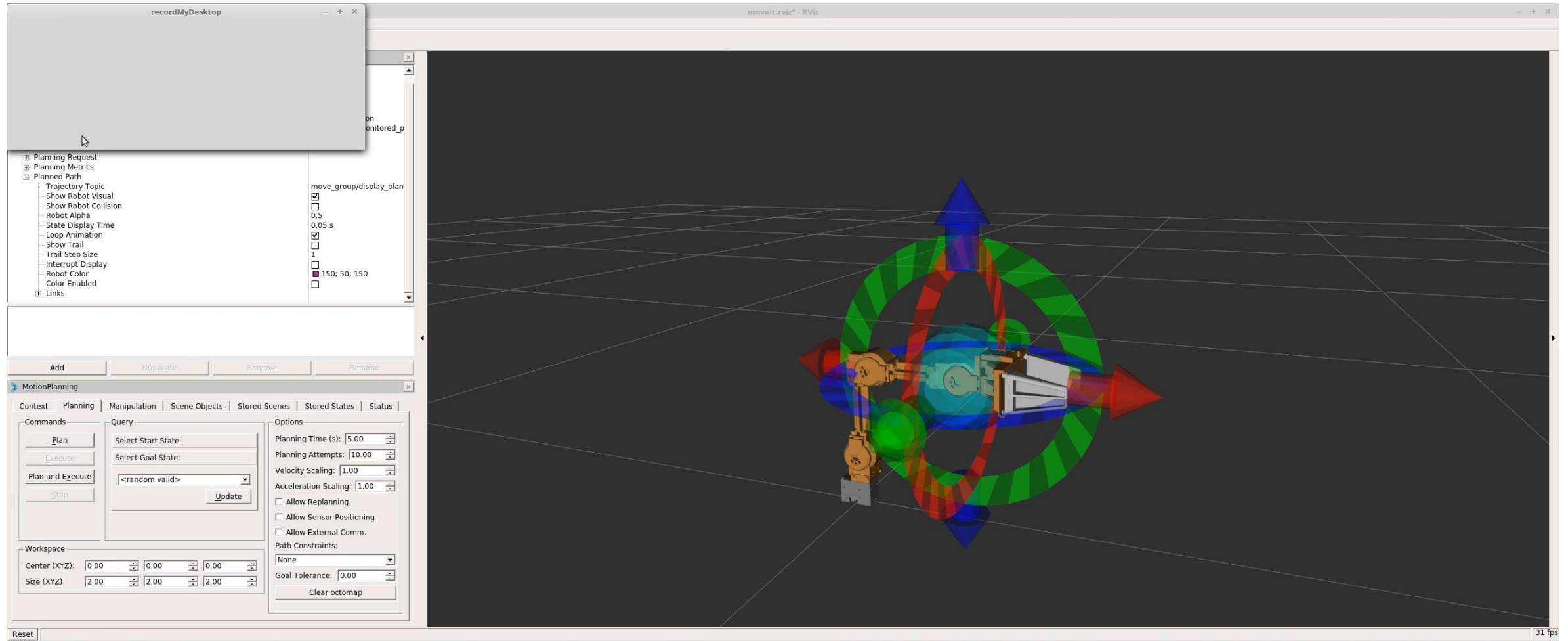


"HIS PATH-PLANNING MAY BE
SUB-OPTIMAL, BUT IT'S GOT FLAIR."

Navigation Demo

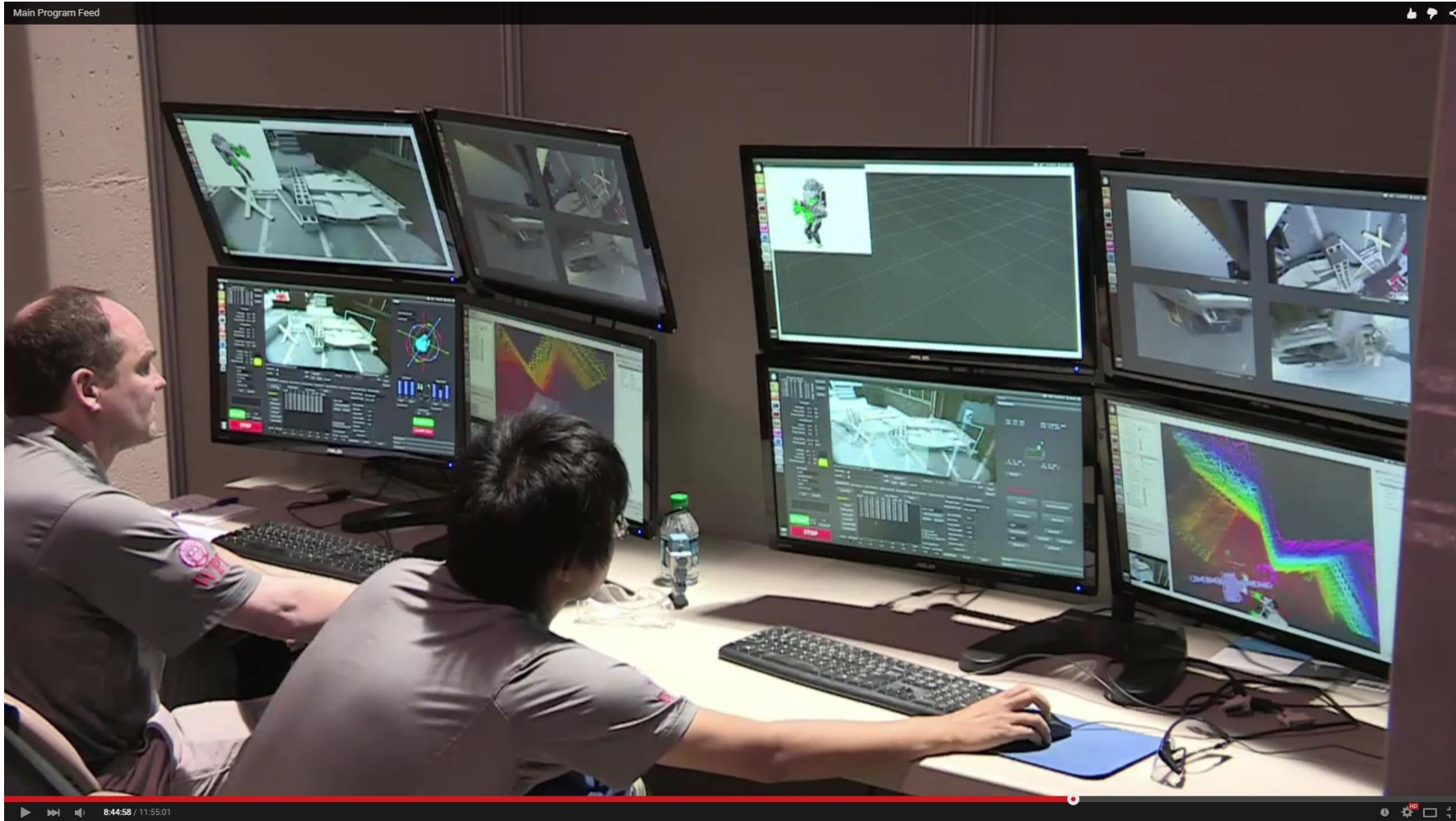
RViz Example #9

- IK target position and path display using interactive markers



RViz Example #10

- Disaster relief robots (2015 DARPA Robotics Challenge)



RViz Installation and Test

- RViz Installation

```
$ sudo apt-get install ros-kinetic-rviz
```

* If you installed 'ros-kinetic-desktop-full', RViz will be installed by default

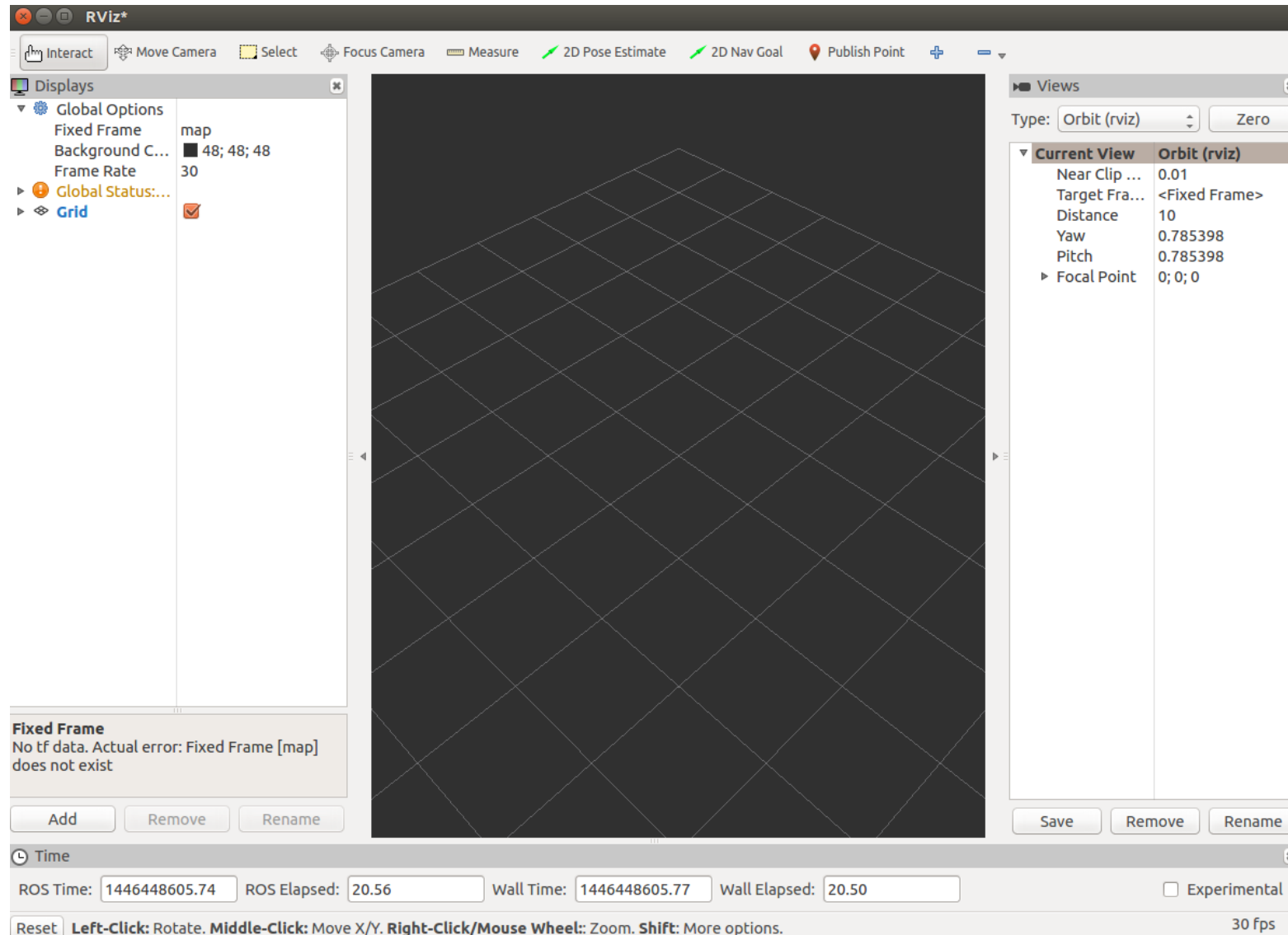
- Run RViz

```
$ rosrun rviz rviz
```

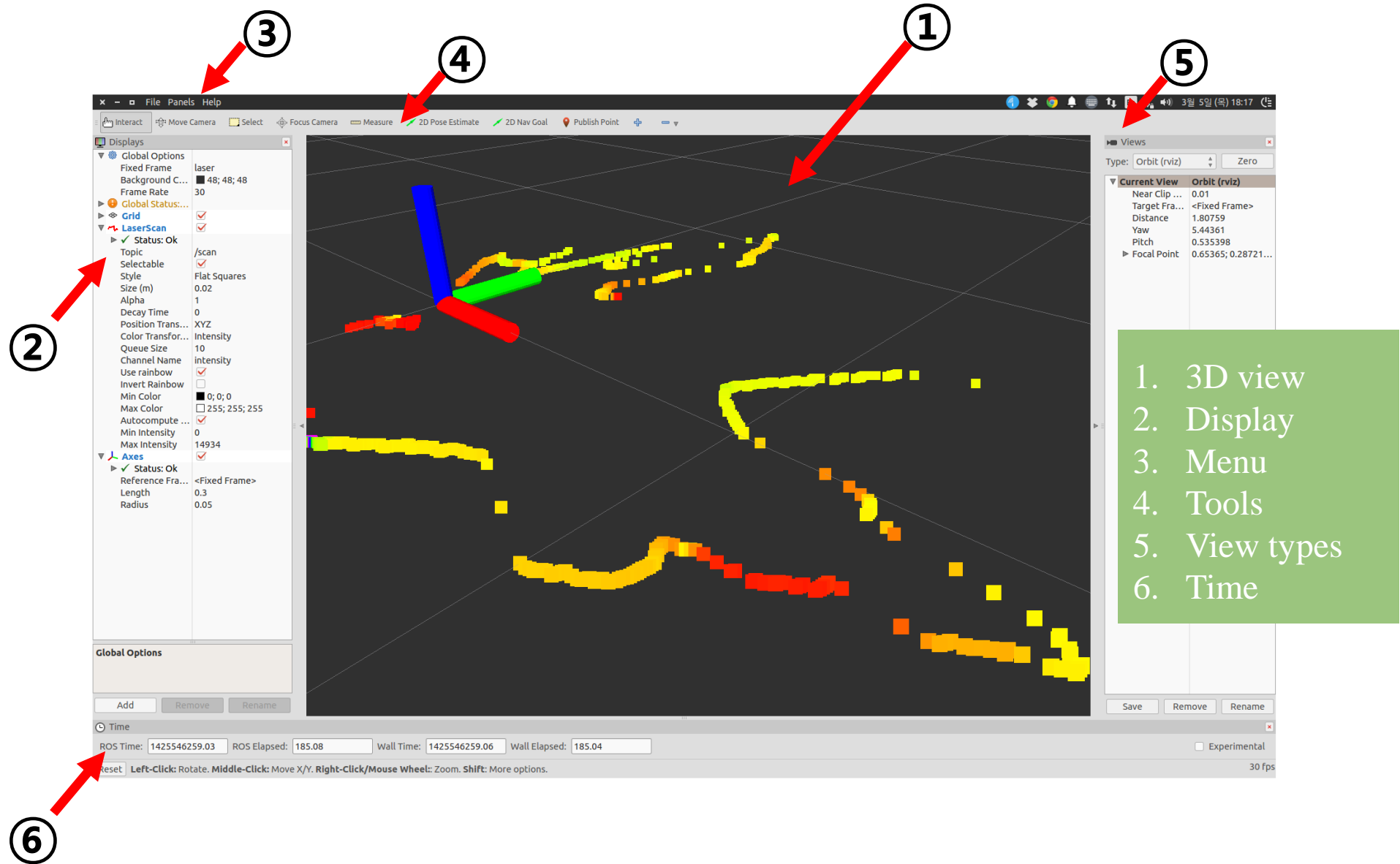
or

```
$ rviz
```

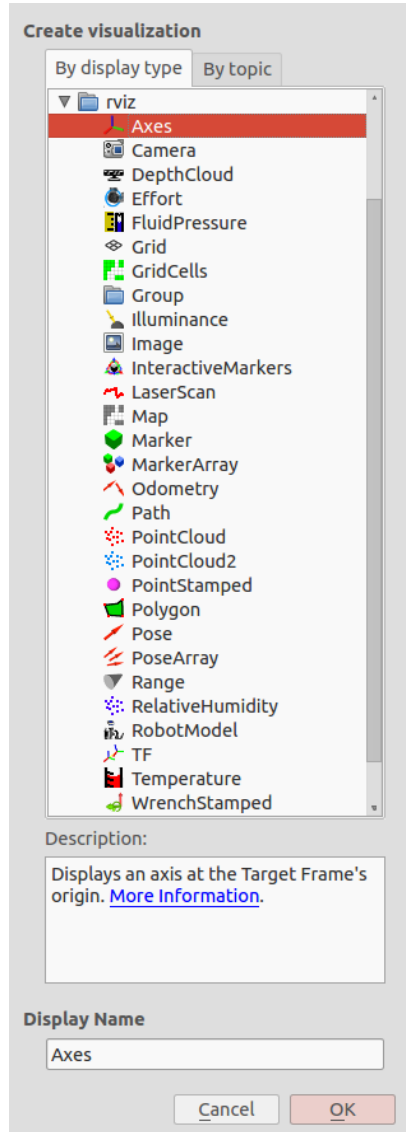

RViz initial screen (not configured yet)
































Screen configuration of RViz (for LDS)



Display type of RViz (Click 'ADD' in Display Menu)



-  Axes
-  Camera
-  Depth cloud
-  Effort
-  Fluid pressure
-  Grid
-  Grid cells (used for map)
-  Group
-  Illuminance
-  Video
-  Interactive marker
-  Laser scan
-  Map
-  Marker
-  Marker array
-  Odometry

-  Path
-  Point cloud
-  Point cloud2
-  Point stamped
-  Polygon
-  Pose
-  Pose array
-  Range
-  Temperature
-  Robot model
-  TF
-  Relative Humidity
-  WrenchStamped

With 'RViz',

Sensor and Robot Related

Data visualization becomes very Simple!

GUI Tool Box: RQT

RQT: Plug-in Type Comprehensive GUI Tool for ROS

- Starting with the ROS Fuerte version, the existing 'rxbag', 'rxplot', 'rxgraph', etc. have been merged with 'rqt'. It is now available as **comprehensive GUI tool** for ROS with plug-ins such as 'rqt_bag', 'rqt_plot', 'rqt_graph', etc.
- Since 'rqt' is developed with 'Qt', users can freely add and develop **plugins**
- Let's take a look at '**rqt_image_view**', '**rqt_graph**', '**rqt_plot**', '**rqt_bag**' which are representative plugins of 'rqt'
- In addition, there are plugins such as
- rqt_action, rqt_gui, rqt_plot, rqt_runtime_monitor, rqt_bag, rqt_gui_cpp, rqt_pose_view, rqt_rviz, rqt_bag_plugins, rqt_gui_py, rqt_publisher, rqt_service_caller, rqt_capabilities, rqt_image_view, rqt_py_common, rqt_shell, rqt_console, rqt_launch, rqt_py_console, rqt_srv, rqt_controller_manager, rqt_logger_level, rqt_reconfigure, rqt_tf_tree, rqt_dep, rqt_moveit, rqt_robot_dashboard, rqt_top, rqt_ez_publisher, rqt_msg, rqt_robot_monitor, rqt_topic, rqt_graph, rqt_nav_view, rqt_robot_steering, rqt_web, etc. (wow.. ——;;)

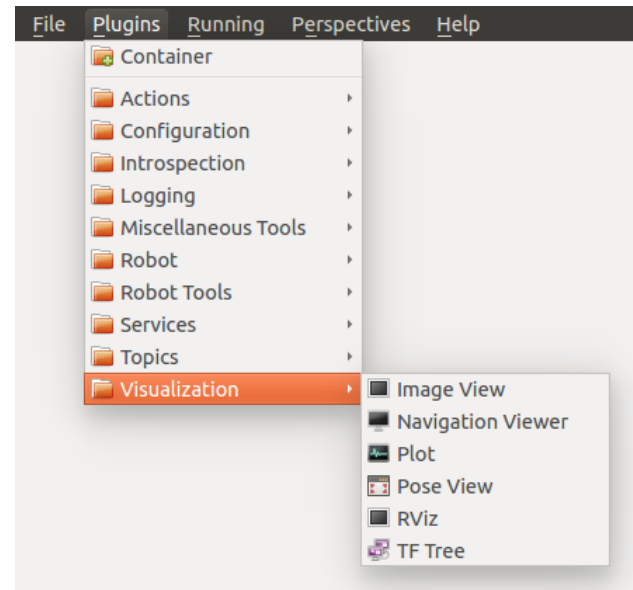
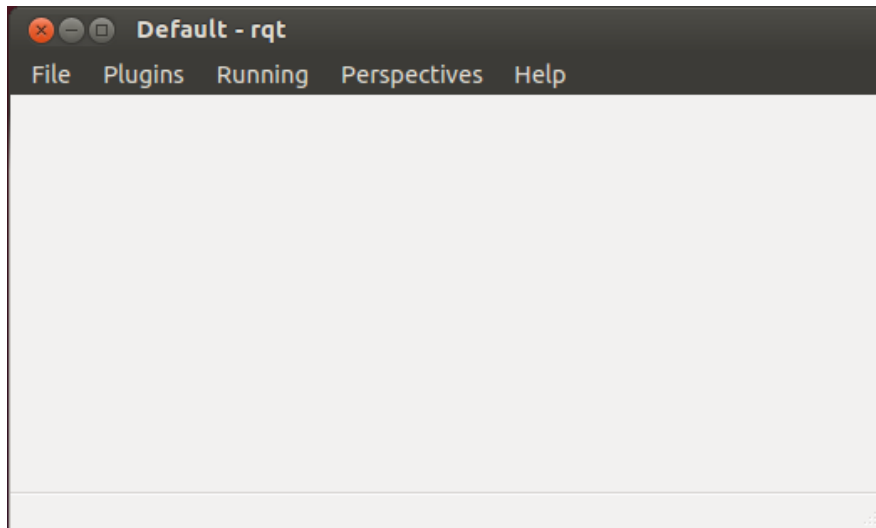
RQT Installation and Test

- RQT Installation

```
$ sudo apt-get install ros-kinetic-rqt ros-kinetic-rqt-common-plugins
```

- RQT Run

```
$ rqt
```



RQT Plug-in #1

1. Action

- **Action Type Browser** | Check the data structure of action type

2. Configuration

- **Dynamic Reconfigure** | Change the GUI setting value to change the setting value provide by the nodes
- **Launch** | GUI version of 'roslaunch'

3. Introspection

- **Node Graph** | Graph view showing relationship diagrams and message flow of running nodes
- **Package Graph** | Graph view showing node dependencies
- **Process Monitor** | Check CPU utilization, memory usage, and number of threads of running nodes

4. Logging

- **Bag** | ROS data logging
- **Console** | Check for messages such as warning, error that occur on the nodes
- **Logger Level** | Select and display logger information such as Debug, Info, Warn, Error, Fatal

RQT Plug-in #2

5. Miscellaneous Tools

- **Python Console** | Python console screen
- **Shell** | Activate shell
- **Web** | Activate web browser

6. Robot

- Depending on the robot, add a plug-in such as a dashboard

7. Robot Tools

- **Controller Manager** | Plug-in required to control the controller
- **Diagnostic Viewer** | Check robot device and error
- **Moveit! Monitor** | Check 'Moveit!' data used in robot arm planning
- **Robot Steering** | Robot adjustment GUI tool, used in remote control to steer the robot
- **Runtime Monitor** | Check for errors and warning on nodes in real time

RQT Plug-in #3

8. Services

- **Service Caller** | Connect to the running service server and request service
- **Service Type Browser** | Check the data structure of the service type

9. Topics

- **Easy Message Publisher** | Publish topic in GUI environment
- **Topic Publisher** | Create and publish topic
- **Topic Type Browser** | Check the data structure of the topic type
- **Topic Monitor** | Check the information of selected topic

10. Visualization

- **Image View** | Check image data of camera
- **Navigation Viewer** | Check location and target point of robot navigation
- **Plot** | 2D data plot GUI plug-in, 2D data plotting
- **Pose View** | Show current TF location and model location
- **RViz** | Rviz plug-in which is 3D visualization tool
- **TF Tree** | Graph view showing tf relation as a tree structure

RQT Example

The screenshot displays the RQT interface with several panels:

- Web Panel:** Shows the ROS.org website with the "Documentation" section selected.
- Topic Publisher Panel:** Displays a table of topics and their expressions.

topic	type	rate	enabled	expression
/cmd_vel2	std_msgs/Float32	10.00	True	$\cos(i/20)*20$
data	float32			
/cmd_vel3	std_msgs/Float32	5.00	True	$\sin(i/20)*10$
data	float32			
- Robot Steering Panel:** Features a slider for the /cmd_vel topic, ranging from -3.00 to 3.00 Hz.
- Logger Level Panel:** Shows a list of nodes and loggers with their respective levels (Debug, Warn, Error, Fatal).
- Console Panel:** Displays a list of messages with their severity, node, and time.

Message	Severity	Node	Time
#9 Loading Setup Assistant Complete	Info	/moveit_setup_assistant	11:11:25.344 (2012-08-02)
#8 Listening to 'moveit_planning_scene'	Info	/moveit_setup_assistant	11:11:25.294 (2012-08-02)
#7 Starting scene monitor	Info	/moveit_setup_assistant	11:11:25.293 (2012-08-02)
#6 Configuring kinematics solvers	Info	/moveit_setup_assistant	11:11:25.107 (2012-08-02)
#4 Robot semantic model successfully loaded.	Info	/moveit_setup_assistant	11:11:23.119 (2012-08-02)
#5 Setting Param Server with Robot Seman...	Info	/moveit_setup_assistant	11:11:23.119 (2012-08-02)
- Plot Panel:** Shows a graph of two sinusoidal signals: /cmd_vel2/data (red line) and /cmd_vel3/data (blue line).

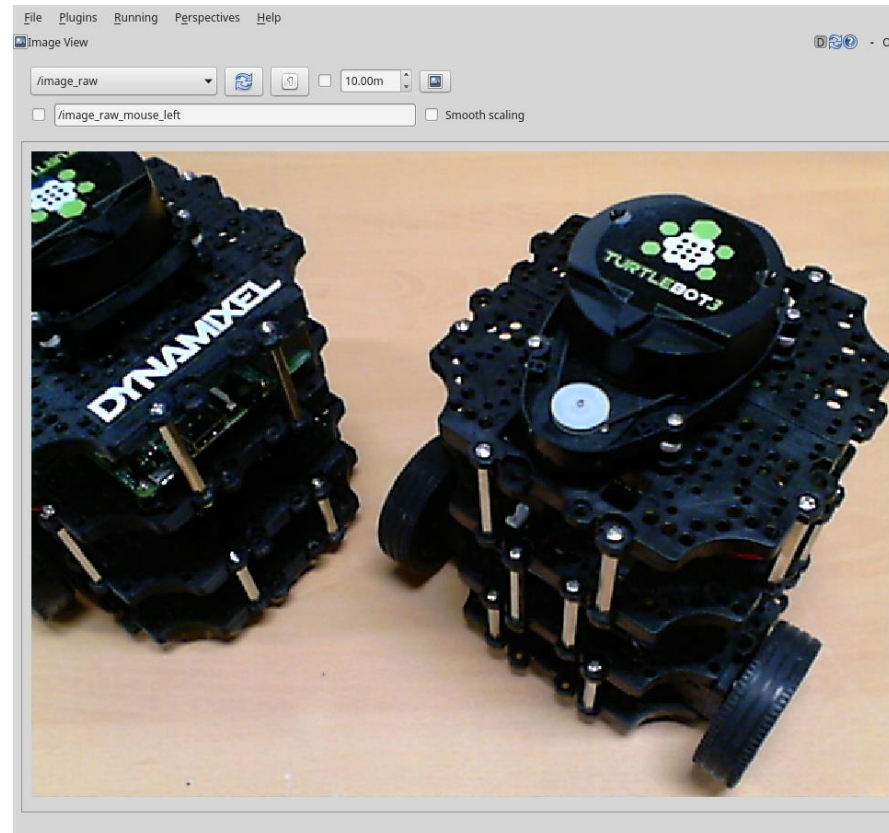
RQT Practice #1: rqt_image_view

```
$ rosrun uvc_camera uvc_camera_node
```

```
$ rqt    ( Select [Plugins] → [Visualization] → [Image View] in menu )
```

or

```
$ rqt_image_view
```



RQT Practice #2: rqt_graph

```
$ rosrun turtlesim turtlesim_node
```

```
$ rosrun turtlesim turtle_teleop_key
```

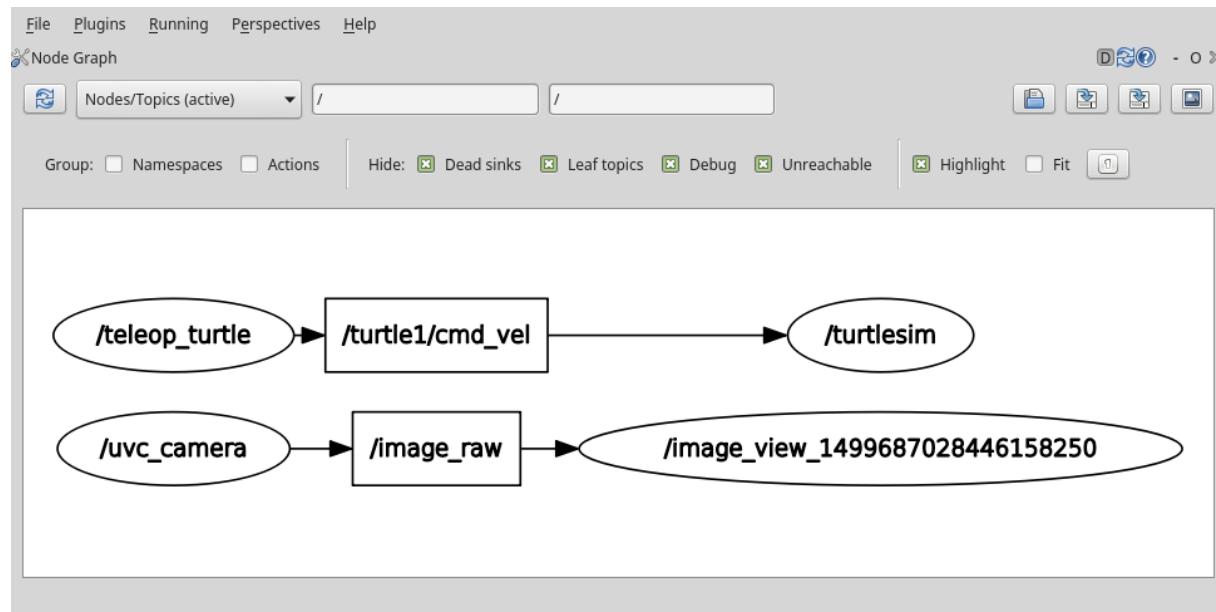
```
$ rosrun uvc_camera uvc_camera_node
```

```
$ rosrun image_view image_view image:=image_raw
```

\$ rqt (Select [Plugins] → [Introspection] → [Node_Graph] in menu)

or

```
$ rqt_graph
```



RQT Practice #3: rqt_plot

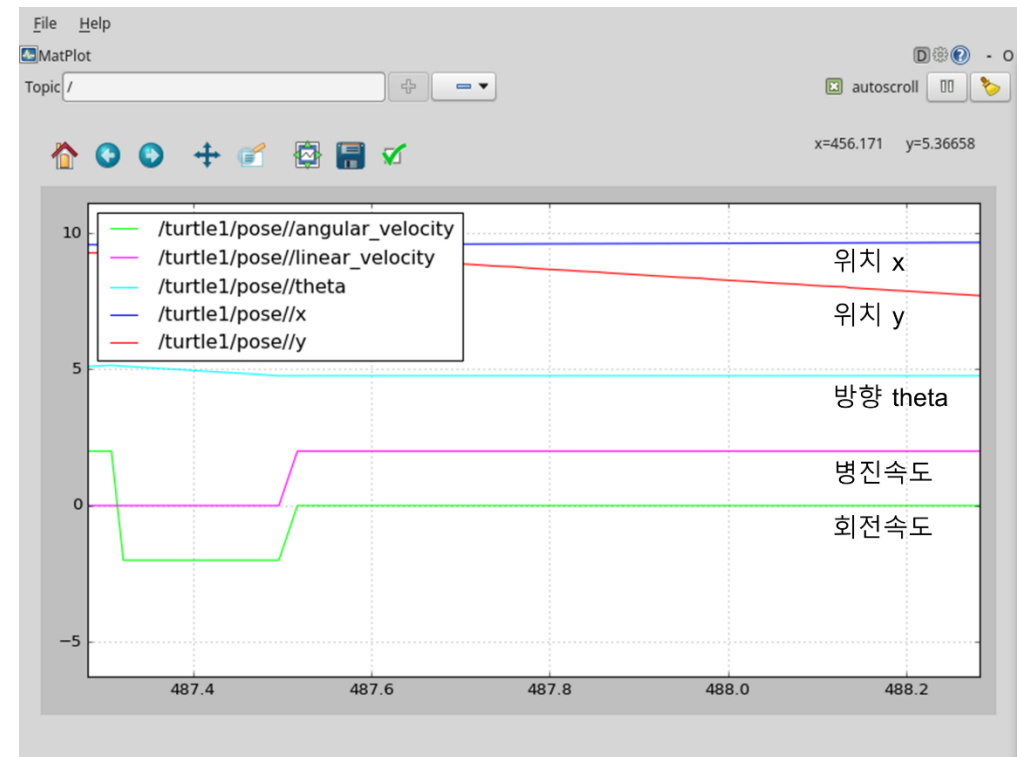
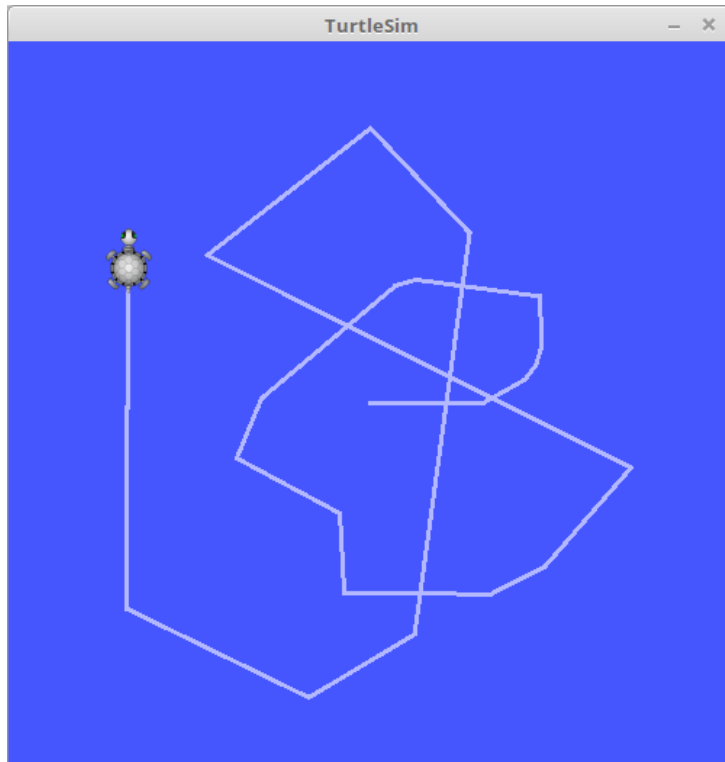
```
$ rosrun turtlesim turtlesim_node
```

```
$ rosrun turtlesim turtle_teleop_key
```

```
$ rqt ( Select [Plugins] → [Visualization] → [Plot] in menu )
```

or

```
$ rqt_plot /turtle1/pose/
```



RQT Practice #4: rqt_bag

```
$ rosrun uvc_camera uvc_camera_node
```

```
$ rosbag record /image_raw
```

```
$ rqt    ( Select [Plugins] → [Logging] → [Bag] in menu )
```

or

```
$ rqt_bag
```



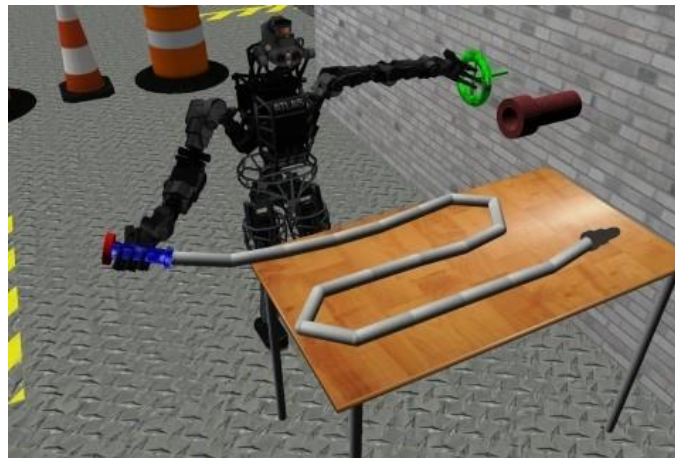
Using RQT

1. ROS available in GUI form
2. Easy to create GUI Tool!

3D Simulator: Gazebo

Gazebo

- Gazebo is 3D Simulator with Physics Engine, Robot model, Sensor, Environment model, and so on. It helps you to get data similar to the one in real environment.
- Gazebo is regarded as the best simulator among recently-introduced Open Simulator. In addition, it is selected as an **official simulator for DARPA Robotics Challenge**
- It is highly compatible with ROS.



Dynamics Simulation

Access multiple high-performance physics engines including [ODE](#), [Bullet](#), [Simbody](#), and [DART](#).



Advanced 3D Graphics

Utilizing [OGRE](#), Gazebo provides realistic rendering of environments including high-quality lighting, shadows, and textures.



Sensors and Noise

Generate sensor data, optionally with noise, from laser range finders, 2D/3D cameras, Kinect style sensors, contact sensors, force-torque, and more.



Plugins

Develop custom plugins for robot, sensor, and environmental control. Plugins provide direct access to Gazebo's [API](#).



Robot Models

Many robots are provided including PR2, Pioneer2 DX, iRobot Create, and TurtleBot. Or build your own using [SDF](#).



TCP/IP Transport

Run simulation on remote servers, and interface to Gazebo through socket-based message passing using Google [Protobufs](#).



Cloud Simulation

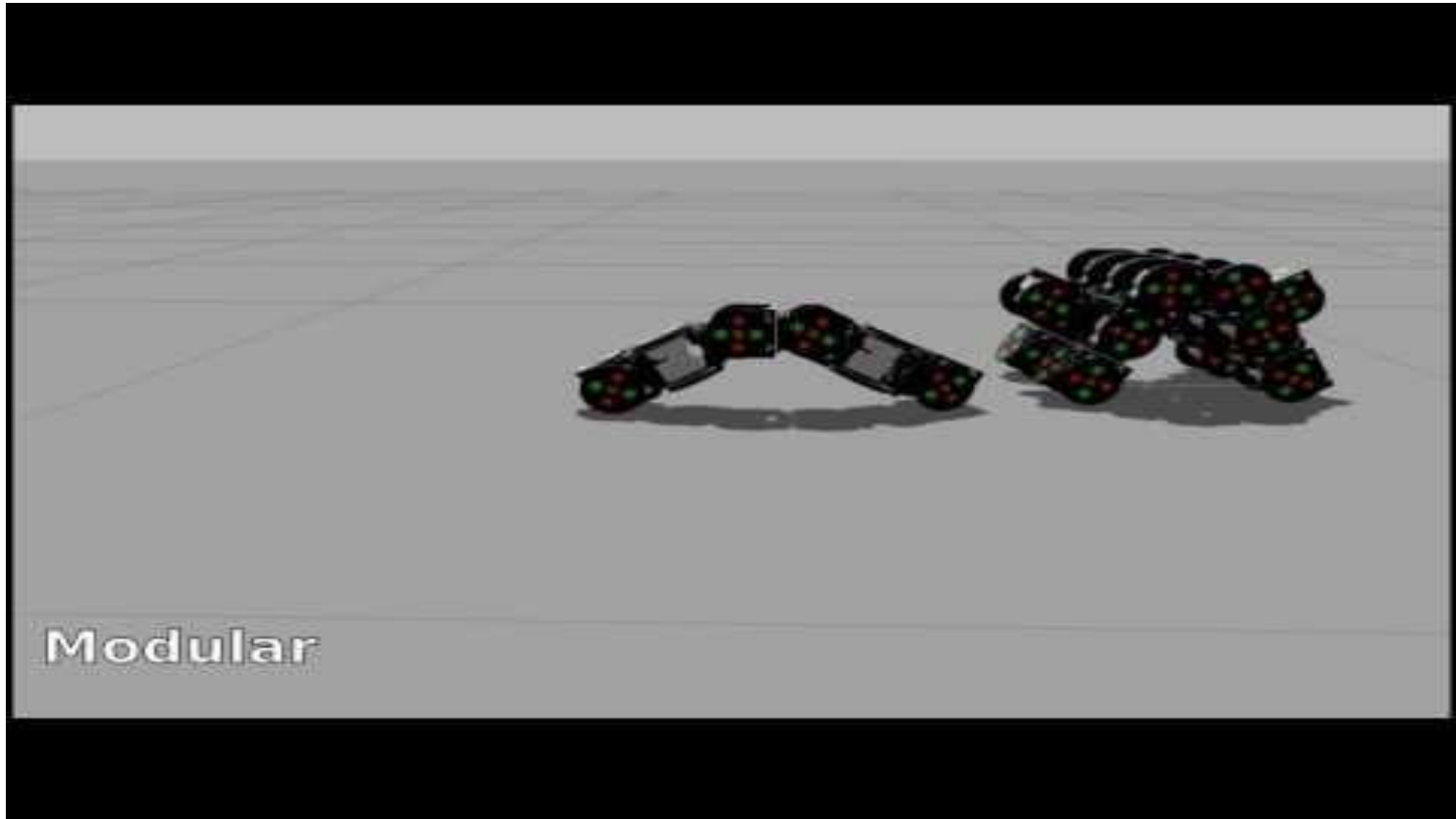
Use [CloudSim](#) to run Gazebo on Amazon, Softlayer, or your own OpenStack instance.



Command Line Tools

Extensive command line tools facilitate simulation introspection and control.

Gazebo



Key Point?

If you need **Simulation**,
It is easy to work with ROS &
Gazebo!!

Question Time!

Advertisement #1



Free

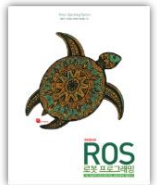


Download link



Language:

English, chinese, Japanese, Korean



“ROS Robot Programming”

A Handbook is written by TurtleBot3 Developers

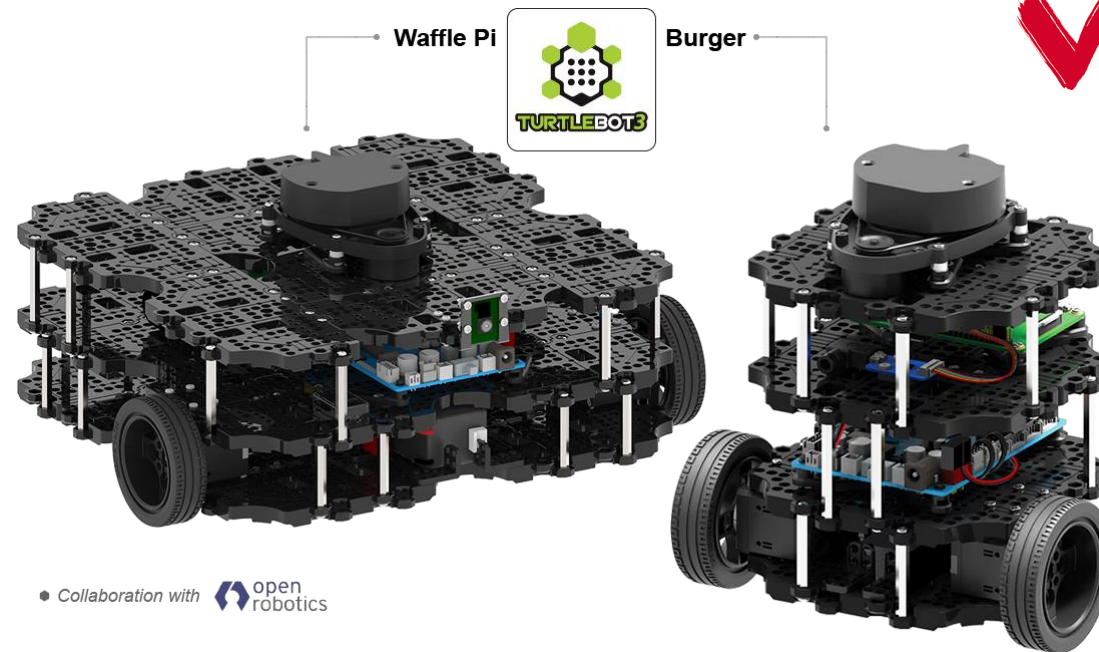
Advertisement #2

TURTLEBOT3

AI Research Starts Here
ROS Official Platform

TurtleBot3 is a new generation mobile robot that's modular, compact and customizable. Let's explore ROS and create exciting applications for education, research and product development.

✓ [Direct Link](#)



• Collaboration with  open robotics

Advertisement #3



www.robotsource.org

The 'RobotSource' community is the space for people making robots.

We hope to be a community where we can share knowledge about robots, share robot development information and experiences, help each other and collaborate together. Through this community, we want to realize open robotics without distinguishing between students, universities, research institutes and companies.

Join us in the Robot community ~

END.