Introduction to ROS

Inteligencia Artificial en los Sistemas de Control Autónomo
Máster Universitario en Ingeniería Industrial

Departamento de Automática

Universidad de Alcalá

ISG

Objectives

- Understand the need of robot abstractions
- Overview the main robotic software development problems
- Introduce the ROS ecosystem
- First contact with ROS

Bibliography

No suiteable bibliography

Table of Contents

Introduction (I)

Complexity of robotic software

- Different robot morphologies
- Different sensors
- Different actuators
- Different CPU/microcontroller

Complex high-level logic

- Computer vision, SLAM, navigation, ...
- Software reuse is a big issue in Robotics

Solution: Abstraction

Introduction (II)

Robotic platforms

- ROS
- Player
- Rock
- Pyro

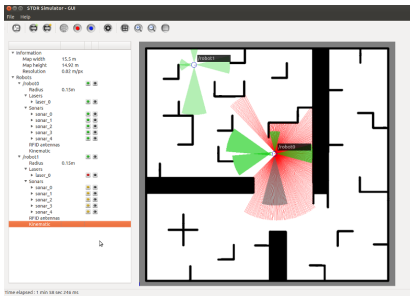
Problem: Testing

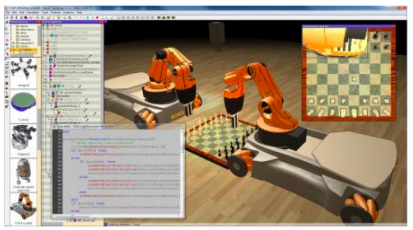
Introduction (III)

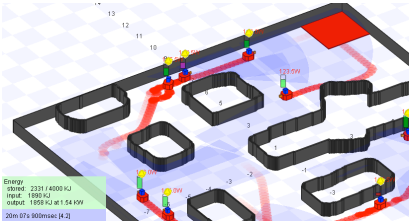
Simulation

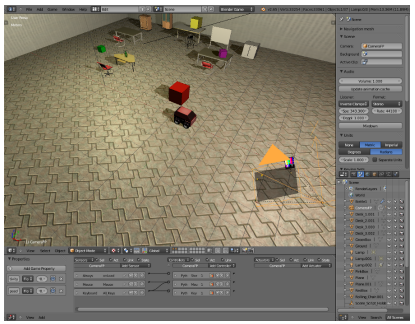
- Gazebo
- STDR
- V-Rep
- Stage
- MORSE
- Webots

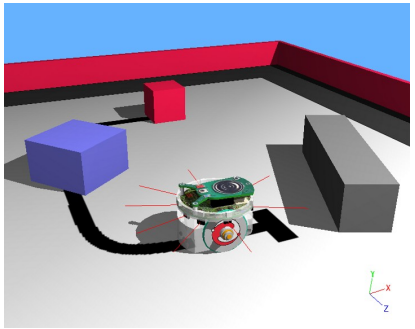
(Video)











Problem: Perception visualization

Introduction (IV)

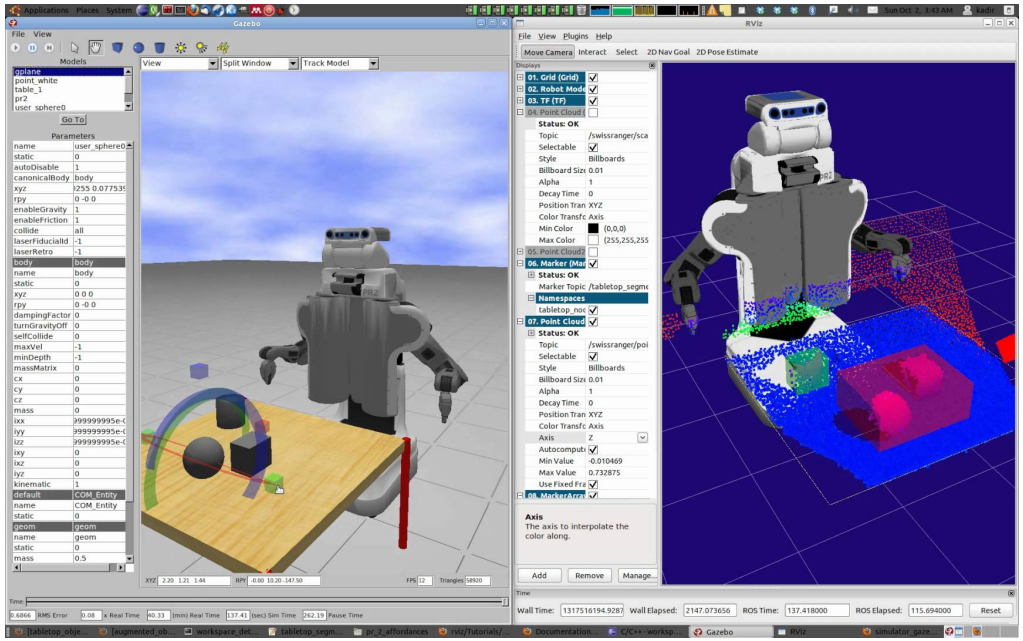
Sensor visualization

- RViz

Complex motion

- MoveIt!

(Video)



Introduction (V)

The robot developer toolkit

- Robot
- Linux ⇒ Operating system
- ROS ⇒ Robotic platform
- Gazebo ⇒ Simulator
- RViz ⇒ Sensor visualization
- MoveIt! ⇒ Motion operation

Robot Operating System (ROS) (I)

ROS is an "operating system" for robots

- Actually, despite its name, ROS is not an operating system
- Located on the operating system and under the applications
- ROS is a *robotic platform*

ROS was created in the Standford AI Lab, in 2007

- Continuous development since then
- Almost a standard in research
- Increasing popularity in the industry

Robot Operating System (ROS) (II)

ROS provides

- An execution environment
- A computation model (robot abstraction)
- Tools (like RViz and MoveIt!)
- Hardware independence

Features

- Robot drivers
- Advanced algorithms (SLAM, navigation, etc)
- Large number of supported sensors
- Pose estimation, localization, and navigation
- Standard robot messages

ROS versions

Version	Release	Logo	
Lunar Loggerhead	May 23rd, 2017		
Kinetic Kame	May 23rd, 2016		
Jade Turtle	May 23rd, 2015		
Indigo Igloo	July 22nd, 2014		Stable
Hydro Medusa	Sept. 4th, 2013		
Groovy Galapagos	Dec. 31, 2012		
Fuerte Turtle	April 23, 2012		
Electric Emys	August 30, 2011		

Installation

ROS runs over Linux

- Ubuntu recommended (official packages available)
- Ubuntu and ROS version strongly coupled
- Any Linux machine supports, in theory, ROS
- Warning: Installing ROS on weird platforms hurts!

Detailed installation instructions available

- `http://wiki.ros.org/jade/installation/`

We will use a VM with ROS installed in advance (you're welcome :)

Documentation

- All the ROS documentation is available on its wiki
- ROS tutorials are a good entry point to ROS
- Be careful with ROS version
- Each robot is coded in a package, each package has its own documentation

Exercises

Environment setup

1. Install VM with ROS Indigo
   - VM image in `http://nootrix.com/software/ros-indigo-virtual-machine/`
   - User: `viki`, password: `viki`
2. Configure Spanish keyboard in the VM
3. Set up the display resolution in the VM
4. Update packages in the VM (do not jump this step!)
5. Install ROS Turtlebot packages
   - `apt-get install ros-indigo-turtlebot-*`
   - `apt-get install ros-indigo-robot-state-publisher`
   - Close terminal (important!)

Exercises

Basic rover: TurtleBot

1. Run Gazebo simulation
   - `roslaunch turtlebot_gazebo turtlebot_world.launch`
2. Run Turtlebot teleoperation
   - `roslaunch turtlebot_teleop keyboard_teleop.launch`
3. Run RViz real-time data visualization
   - `roslaunch turtlebot_rviz_launchers view_robot.launch`

Exercises

Mobile manipulator: PR2

1. Install PR2 and Moveit!
   - `sudo apt-get install ros-indigo-moveit-pr2 ros-indigo-moveit-ros-visualization`
   - Close the terminal and open a new one
2. Run RViz and Moveit
   - `roslaunch pr2_moveit_config demo.launch`

Exercises

UAV: Hector quadrotor

1. Install Hector quadrotor:
   - `sudo apt-get install ros-indigo-hector-*`
   - Close the terminal and open a new one
2. Run simulation
   - `roslaunch hector_quadrotor_demo outdoor_flight_gazebo.launch`
3. Run teleoperation (with joystick)
   - `roslaunch hector_quadrotor_teleop xbox_controller.launch`

Exercises

UAV: Hector quadrotor (indoor SLAM)

1. Run simulation
   - `roslaunch hector_quadrotor_demo indoor_slam_gazebo.launch`
2. Run teleoperation (with joystick)
   - `roslaunch hector_quadrotor_teleop xbox_controller.launch`

Exercises

Humanoid: Nao

1. Install Nao packages
   - `sudo apt-get install ros-indigo-nao*`
2. Run Nao demo
   - `roslaunch nao_gazebo_plugin nao_gazebo_plugin_H25.launch`
3. Move robot with Moveit
   - `roslaunch nao_moveit_config moveit_planner.launch`