Objectives

- Introduce ROS
- Present ROS concepts and keywords
- Present some ROS interfaces
- Show examples of robots covered by ROS
Outline

- What is ROS?
- Where can I run ROS
- What set ROS apart?
- How is ROS organized
- How do nodes Talk
- ROS client libraries
- Command-line tools
- GUI tools for ROS
- ROS wiki
- Who’s using ROS
- How flexible is ROS
- What is inside?
- Summary
- References
What is ROS?

An operating system for robotics that provides:

- Hardware abstraction
- Low-level device control
- Implementation of commonly-used functionalities
- Message-passing between processes
- Package management
What is ROS?

• 2007: Originally developed by the Stanford Artificial Intelligence Laboratory

• Since 2008: Development continues at Willow Garage, a robotics research institute

Philosophical goals of ROS:

• Peer-to-peer
• Tools-based
• Multi-lingual
• Free and Open-Source
Where can I run ROS?

Supported
- Ubuntu

Experimental
- Arch
- Mac OS X
- Fedora

Partial Functionality
- Gentoo
- OpenSUSE
- Slackware
- Windows
- FreeBSD
What sets ROS apart?

Modular

Multi-language

Tools Based

Open Source
Modular
How is ROS organized?

- **Packages**
  - Nodes
  - Messages
  - Services

- **Stacks**
  
  Groups of packages

- **Modular**
How do Nodes talk?

Messages

Node A

Topic

Node B

Modular
How do Nodes talk?

Services

Node A

Request

Response

Node B

Modular
How do Nodes talk?

Actions

Goal
Cancel
Status
Result
Feedback

Action Client

Action Server

Modular
Multi-language
ROS Client Libraries

ROS client libraries allow nodes written in different programming languages to communicate

- roscpp - C++ client library
- rosyp - Python client library

Experimental Client Libraries

- rosoct - Octave client library
- roslip - LISP client library
- rosjava - Java client library
- roslua - Lua client library
Tools Based
Command-line Tools

Philosophy: Various tools instead of a runtime environment

- rosbag
- rosbash
- roscd
- roscleanup
- roscore
- rosdep
- rosed
- rosrun
- roscreate-pkg
- roscreate-stack
- rostopic
- roslauch
- roslocate
- rosmake
- rosmsg
- rosnodel
- rospack
- rosparam
- rossrv
- rosservice
- rossstack
GUI Tools for ROS

rx Stack

- rxbag - ROS bag recording, playback and visualization
- rxgraph - ROS graph visualization
- rxplot - ROS topic data visualization
- rxconsole - ROS logging output

rviz

- A 3D visualization environment for robots using ROS

Tools Based
rosbag and rxbag
rviz

Tools Based
Open Source
ROS Wiki
ROS Wiki

• Free for commercial and research use
• Installation Instructions
• Documentation
• Tutorials
• Community support
• You can contribute too!

Open Source
Who’s using ROS?

Over 120 ROS repositories worldwide

Open Source
Robots
How flexible is ROS?

Robots

- PR2
- Marvin Autonomous Car
- Aldebaran Nao
- Shadow Robot
- Robotino
- Roomba / Create
- AscTec Quadrotor
- Lego NXT
- Care-O-Bot 3
How flexible is ROS?

Microsoft Kinect

Hokuyo Laser

Sick Laser

Xsens Mti IMU

3D PMD Camcube

Mercury5e RFID Reader

Arduino Integration

BMA Accelerometer

Robots

ROS Workshop
What’s inside?
What’s inside?

• User contributed packages that implement functionalities such as SLAM, navigation, etc.

• Support for many processors running distributedly

• ROS places virtually all complexity in libraries, only creating small executables which expose library functionalities to ROS

• ROS allows the re-uses of code from numerous other open-source projects such as:
  – Drivers, navigation system and simulators from Player/Stage
  – Vision algorithms from OpenCV
  – Planning algorithms from OpenRAVE
Visual SLAM
Thank you!

Special Thanks to David Portugal and Gonçalo Cabrita (ISR – COIMBRA)