ROS for devices and robots

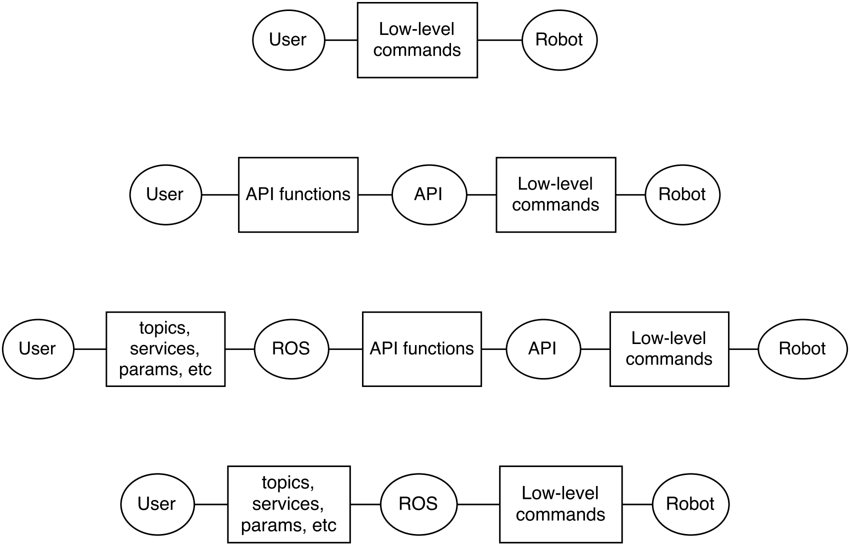
Some general theory and real examples

Department of Control Systems and Informatics
ITMO University, Saint Petersburg, Russia

Evgeniy S. Antonov

20 February, 2017

Different schemes of user–robot interaction



Some simple examples

joy and *teleop_twist_joy* packages

joy package:

- joy package allows to work with standard joysticks.
- Installation—building from sources or by command

```
$ sudo apt-get install ros-"distro"-joy
```

teleop_twist_joy package:

- teleop_twist_joy package contains a node which translates commands from joystick into velocity commands.
- Installation—building from sources or by command

```
$ sudo apt-get install ros-"distro"-teleop-twist-joy
```

Example of their using:

- Look at this launch file.

Some simple examples

usb_cam package

- usb_cam package allows to work with standard webcams.
- Installation—
  - for indigo and jade:

    ```
    $ sudo apt-get install ros-"distro"-usb-cam
    ```

    or just like for kinetic
  - for kinetic:

    ```
    $ cd PATH_TO_YOUR_WORKSPACE/src
    $ git clone https://github.com/bosch-ros-pkg/usb_cam.git
    $ cd ..
    $ catkin_make --pkg usb_cam
    ```
- For trying this package in work use its own launch:

  ```
  $ roslaunch usb_cam usb_cam-test.launch
  ```

Some simple examples

image_view package

- Last mentioned launch file also runs an image_view node from image_view package.
- The latter is used for showing webcam's image in a separate window.
- Its instalation—building from sources or by command

```
$ sudo apt-get install ros-"distro"-image-view
```

Robots' 3D models

urdf package

- The urdf package creates a robot's 3D model based on its URDF description (Unified Robot Description Format).
- It is highly recommended to read[1]
    - this package's description
    - its tutorials
    - *xacro* package's description

[1] Don't panic due to these materials' complexity!

Robots' 3D models

robot_state_publisher package

- The robot_state_publisher allows to publish the state of a robot to *tf* using robot's URDF description and a special topic which must be provided by robot's driver.
- It is highly recommended to read
  - this package's description
  - its tutorials

Robots' 3D models

joint_state_publisher package

- The joint_state_publisher allows to publish fake data for robot joints' states using robot's URDF description.
- It is highly recommended to read
  - this package's description

Robotino

Related resources

- Its wiki page (it is not very necessary to read).
- Its manual in which you can find information about charging of the robot and places of its I/O ports.
- A document about connection of the gripper.
- New Robotino's manual. Refer to it for rules about connection of optical and inductive sensors.
- A description of some of *robotino_node* topics and parameters.

Robotino

Installation of required software

- install robot's standard library (method 1):
  - add debprepository for robot's standard library as it is described here
  - install robot's standard library:

    ```
    $ sudo apt-get update
    $ sudo apt-get install robotino-api2
    ```

- install robot's standard library (method 2, alternative):
  - build it from source almost as described here (XX = 32 or 64):

    ```
    $ cd ANY_DIR
    $ svn co svn://svn.rec.de/openrobotino/api2/trunk source/api2
    $ find source/api2/external -wholename "*/bin/*"\
      -exec chmod +x {} \;
    $ mkdir -p build/api2
    $ cd build/api2
    $ cmake ../../source/api2
    $ make install
    $ echo "ROBOTINOAPI2_XX_DIR=/home/robot/build/install\
      /usr/local/robotino/api2/" >> ~/.bashrc
    $ source ~/.bashrc
    $ ./create_package.sh
    $ sudo dpkg -i robotino-api*.deb
    ```

Robotino

Installation of required software

- copy packages *robotino_description*, *robotino_msgs* and *robotino_node* from here into your workspace
- build them:

```
$ cd PATH_TO_YOUR_WORKSPACE
$ catkin_make --pkg robotino_msgs
$ catkin_make
```

Robotino

Launching

1. If Robotino is connected to public network (CSI-Robotics), then go to step 2. In other case connect your PC to Robotino's network (e.g. Robotino2).

2. launch an appropriate file from *robotino_node* package (don't forget specify robot's IP in it):
   - *only_robotino_node.launch*—for launch Robotino without webcams, lidar[2] and odometry
   - *robotino_simple_node.launch*—for launch Robotino without webcams and lidar
   - *robotino_node.launch*—for launch Robotino with webcams, lidar and odometry

[2] It must be connected to the robot using all three ends of its Y-USB cable.

KUKA Youbot

- Its wiki page (it is well writen and must be read before using the robot).

- It is controlled by internal[3] or external PC on which you have installed ROS and these packages: *youbot_description*, *youbot_driver* and *youbot_driver_ros_interface*.

[3] It is managed by Ubuntu 14.04 and ROS Indigo Igloo.

KUKA Youbot

Installation of required software

- For Indigo:
  - install youbot_description package:

    ```
    $ sudo apt-get install ros-indigo-youbot-description
    ```

  - install brics_actuator package (if it is not installed):

    ```
    $ sudo apt-get install ros-indigo-brics-actuator
    ```

- For Jade and Kinetic:
  - install *youbot_description* package:

    ```
    $ cd PATH_TO_YOUR_WORKSPACE/src
    $ git clone https://github.com/youbot/youbot_description.git
    $ cd youbot_description
    $ git checkout -b jade-devel origin/jade-devel
    $ cd ../..
    $ catkin_make --pkg youbot_description
    ```

  - install *brics_actuator* package (if it is not installed):

    ```
    $ cd PATH_TO_YOUR_WORKSPACE/src
    $ git clone https://github.com/wnowak/brics_actuator.git
    $ cd ..
    $ catkin_make --pkg brics_actuator
    ```

KUKA Youbot

Installation of required software

- Common final steps:
  - install pr2_msgs package (if it is not installed):

    ```
    $ sudo apt-get install ros-"distro"-pr2-msgs
    ```

  - install *youbot_driver* and *youbot_driver_ros_interface* packages as it is described in this document

Parrot AR.Drone 2.0[4]

- Its driver package can be installed by command:

  ```
  $ sudo apt-get install ros-"distro"-ardrone-autonomy
  ```

  or by building from its sources.
- Package's documentation lives on a separate site.
- For your first fly
  - connect your computer to drone's wifi
  - specify drone's IPv4 in ardrone.launch:

    ```
    $ rosed ardrone_autonomy ardrone.launch
    ```
  - launch the latter:

    ```
    $ roslaunch ardrone_autonomy ardrone.launch
    ```

[4] Be careful with this robot due to small employees' experience of using it.

Thank you for attention!
And sorry for possible mistakes!