The Service Robot Justina
User's Manual

People from Biorobotics Lab

Facultad de Ingeniería, UNAM

Contents

1 Introduction

This manual gives a general description of the subsystems of the service robot Justina. It is not intended to be a detailed technical report nor a deep explanation of the theoretical basis of Justina's software. The goal of this manual is to give the user a guide for bringing up the robot and solving the most common problems. For each module it is included a brief description of the algorithms, techniques or approaches used for the design, neverthless, bibliography and references are given for the reader interested in a more advanced explanation.

The service robot Justina was designed at the Biorobotics Lab, Faculty of Engineering, UNAM.

The source code of this manual can be found in the folder `JUSTINA/user_manual`.

2 Main Guidelines for Code Developing

- All source code MUST be contained in the folder `catkin_ws/src`.
- Only the code contained in the folder `catkin_ws/src/hardware` can interact with the robot's hardware.
- The previous point implies that all other programs should implement ONLY algorithms. All interaction with the hardware (e.g., getting an image from camera, reading laser, moving base or head, speaking, etc) must be done by interchanging info with the packages contained in the `hardware` folder, through ROS topics and services.
- The code contained in ALL folders inside `catkin_ws/src`, except the `tools` folder, MUST contain only code written by the own developer (of each package). All necessary libraries or code from other sources (serial libraries, arduino, julius, dynamixel, etc), if it is not installed in some default path (`/opt/ros`, `/usr/local/`, etc), then must be put inside the folder `catkin_ws/src/tools` in an appropiate subfolder.
- Developers should try to use only messages already defined in some ros package or stack, neverthless, if custom messages are needed, these must be put inside the `catkin_ws/src/subsystem/subsystem_msgs`, so that, such messages can be used without having to launch all the other subsystems.

3 Folder Structure

```
catkin_ws
    build
    devel
    src
        hardware
            arms
            battery
            hardware_state
            justina_urdf
            hardware_msgs
            head
            mobile_base
            point_cloud_manager
            speakers
            torso
        hri
            gesture_recog
            hri_msgs
            justina_gui
            natural_language
            speech_recog
        interoperation
            bbros_bridge
            joy_teleop
            pc_teleop
            roah_rsbb
        manipulation
            arms_predef_movs
            arms_path_planning
            arms_trajectory_planning
            head_predef_movs
            head_tracking_point
            manipulation_msgs
        navigation
            localization
            mapping
            moving
            navigation_msgs
            path_planning
            point_traking
        planning
            planning_msgs
            pomdp
            rule_based
            semantic_database
            state_machines
        surge_et_ambula
            launch
            rviz_files
        testing
            any_not_stable_node
        tools
```

```
            ros_tools
            libraries
                serial_arduino
                serial_dynamixel
                julius
                festival
        vision
            door_detector
            furniture_recog
            object_detector
            object_recog
            person_detection
            person_recog
            vision_msgs
user_manual
```

Each package in the `hardware` folder should have its simulated version, so that, the rest of the software (all other folders are suposed to contain only algorithms and no interaction with robot's hardware) can run independently of the simulation mode. Choosing between simulated or real should be done in the launch files.

`tools/ros_tools` is intended to have header files or ros packages whose goal is to make easier the interaction with ros, e.g., functions for reading/wrigting ros params, nodes or headers for recording bags.

The following is needed:

- Ubuntu 14.04.1 (This is the tested version)
- ROS Indigo desktop full
- OpenCV 2.4.8 or 2.4.9. Compiled with OpenNi, WITHOUT OpenCL, WITHOUT Cuda, with Eigen
- PCL 1.6

For installing OpenCV 2.4.9:

```
sudo apt get update
sudo apt-get install build-essential libgtk2.0-dev libjpeg-dev libtiff4-dev
     libjasper-dev libopenexr-dev cmake python-dev python-numpy python-tk
     libtbb-dev libeigen3-dev yasm libfaac-dev libopencore-amrnb-dev
     libopencore-amrwb-dev libtheora-dev libvorbis-dev libxvidcore-dev
     libx264-dev libqt4-dev libqt4-opengl-dev sphinx-common texlive-latex-extra
     libv4l-dev libdc1394-22-dev libavcodec-dev libavformat-dev libswscale-dev
     default-jdk ant libvtk5-qt4-dev
cd ~
wget http://sourceforge.net/projects/opencvlibrary/files/opencv-unix/2.4.9/opencv-2.4.9.zip
unzip opencv-2.4.9.zip
cd opencv-2.4.9
mkdir build
cd build
cmake -D WITH_TBB=ON -D BUILD_NEW_PYTHON_SUPPORT=ON -D WITH_V4L=ON -D INSTALL_C_EXAMPLES=ON
     -D INSTALL_PYTHON_EXAMPLES=ON -D BUILD_EXAMPLES=ON -D WITH_QT=ON -D WITH_OPENGL=ON
     -D WITH_VTK=ON -D WITH_OPENNI=ON -D WITH_OPENCL=OFF ..
make
sudo make install
sudo echo "/usr/local/lib" >> /etc/ld.so.conf.d/opencv.conf
sudo ldconfig
```

- sudo apt-get install ros-indigo-amcl

- sudo apt-get install ros-indigo-tf2-bullet
- sudo apt-get install ros-indigo-fake-localization
- sudo apt-get install ros-indigo-map-server
- sudo apt-get install ros-indigo-sound-play
- sudo apt-get install ros-indigo-pocketsphinx

4 How to make a simple planner

La forma más sencilla de aprender a usar a Justina es hacer una pequeña máquina de estados. La mayoría de los sensores pueden ser leídos mediante la suscripción a un tópico y la mayoría de los actuadores pueden ser utilizados publicando el tópico correspondiente. Por ejemplo, para mover la cabeza, es necesario publicar el tópico `/hardware/head/goal_pose` y para leer la posición actual, es necesario susbribirse al tópico `/hardware/head/current_pose`.

Para facilitar