Report Ros Package BlueROV2 and Antenna System

Created by Antonio Camposeo - Polytechnic of Bari

January 2024

1 How install ROS Noetic on Ubuntu 20.04

Source installation: https://wiki.ros.org/noetic/Installation/Ubuntu
Source ROS Tutorials : https://wiki.ros.org/ROS/Tutorials
Open the terminal and follow these commands:

```
#  Setup your sources.list
#  Setup your computer to accept software from packages.ros.org.

sudo sh -c 'echo "deb http://packages.ros.org/ros/ubuntu
$(lsb_release -sc) main" > /etc/apt/sources.list.d/ros-latest.list'

#  Set up your keys
sudo apt install curl # if you haven't already installed curl
curl -s https://raw.githubusercontent.com/ros/rosdistro/master/ros.asc | sudo apt-key add -

#  Installation
sudo apt update

sudo apt install ros-noetic-desktop-full

#  Environment setup
source /opt/ros/noetic/setup.bash
echo "source /opt/ros/noetic/setup.bash" >> ~/.bashrc
source ~/.bashrc
```

1.1 Create a workspace for catkin

```
source /opt/ros/noetic/setup.bash
```

Let's create and build a catkin workspace:

```
mkdir -p ~/catkin_ws/src
cd ~/catkin_ws/
catkin_make
```

1.2 Add BlueROV2 Project

Once catkin workspace has been created download the file of the project from this link and move the folder called bluerov2 into the directory catkin/src, and then go to the catkin packages and run the command *catkin_make*.
Another option is to follow these command.

```
#  install git
sudo apt-get update
sudo apt-get install git -y
```

```
#  install bluerov2 folder from git
cd ~/catkin_ws/src
git clone https://github.com/antoniocamposeo/Modelling-Control-
and-Implementation-of-a-Haptic-System-for-an-Underwater-Robot
rm -d  code_ArduinoUno
rm -d  code_OpenCM904
rm MasterThesis.pdf
rm README.md
cd ..
catkin_make
```

2 BlueRov2 Nodes

2.1 File : *bluerov2_node.py*

Publish Topics:

- */bluerov2/battery*: BatteryState message;
- */bluerov2/state*: State message folder _bluerov2_msg;
- */bluerov2/odometry*: Odometry message, get data or from mavlink or from camera system (to be implemented).

Subscribe Topics:

- */bluerov2/cmd_vel*: Twist message, contain input signal to move the rov.

2.2 File : *bluerov2_controller.py*

Publish Topics:

- */bluerov2/cmd_vel*: Twist messsage, generate the control input based on keyboard control

Subscribe Topics:

- */bluerov2/battery*
- */bluerov2/state*
- */bluerov2/odometry*
- */keyboard_teleop* : Twist message, from keyboard control input.

2.3 File : *bluerov2_teleop.py*

Publish Topics:

- */keyboard_teleop*: Twist massage, from keyboard control input.

3 BlueRov2 Launch Files

3.1 *bluerov2.launch*

This launch file run two nodes, *bluerov2_node.py* and *bluero2_controller.py*

3.2 *bluerov2_teleop_keyboard.launch*

This launch file run the keyboard teleop control, *bluerov2_teleop.py*. The control is related not to the velocity but to the input signal to set for the linear and agular movement of the rov.

3.3 *key_board_velocity.launch*

This launch file is related to the velocity control. Take the PID parameters from the .yaml file and run the velocity control loop based on the specified topics (see the remap tag). The velocity reference is provided by the keyboard.

4 OpenCM904 Nodes

4.1 File : *opencm904_node.py*

Publish Topics:

- */opencm904/motor_actual_position_raw*: Int32MultiArray message, publish data of angular position related the two motors;
- */opencm904/sensor_ft_data*: FTSensor message, publish data related to the force-torque sensor;
- */opencm904/board_status_reset*: Int32MultiArray message, publish data about the status of the board if is connected or not.

Subscribe Topics:

- */opencm904/data_opencm*: String message, get a string with raw data of motors encoder and 6 channels of ft sensors.

4.2 File : *opencm904_control.py*

Publish Topics:

- */opencm904/motor_position*:Int32MultiArray message, publish data of desired motor position

Subscribe Topics:

- */opencm904/motor_actual_position_raw*
- */opencm904/sensor_ft_data*
- */opencm904/board_status_reset*
- */opencm904/control*: Twist message, contain keyboard input signal to move the two motors.

4.3 File : *opnecm904_teleop.py*

Publish Topics:

- */opencm904/control* Twist message, contain keyboard input signal to move the two motors.

4.4 File : *identification_motors_test.py*

Python code to perform the identification task. This script publishes on topic "/motor_position" the input signal generated in the .csv file created by Matlab. The .csv file is taken from the "csv_files" folder.

5 OpenCM904 Launch Files

5.1 *opencm904.launch*

This launch file runs the rosserial node to connect to the opencm board via a serial connection. Then run openc904_node.py and opencm904_control.py. The opencm904_teleop.py node can be active if keyboard control is required.

5.2 *record.launch*

This launch file record the topic that are necessary to be saved in a file .bag. These files are saved in the .ros folder and to convert them to cvs or txt, it is necessary to use some python script or Plottjuggler.

5.3 *identification.launch*

This launch file runs the identification node and start the recording of data.

6 Command List

```
#  Adding environment variables to your path to allow ROS to function.
#  Use when open new terminal to use ros
source ~/catkin_ws/devel/setup.bash

#  Start roscore
roscore

#  Start node opencm904_node.py
rosrun bluerov2 open904_node.py
#  Start node opencm904_control.py
rosrun bluerov2 open904_control.py
#  Start node opencm904_teleop.py
rosrun bluerov2 open904_teleop.py
#  Start node identification_motors_test.py
rosrun bluerov2 identification_motors_test.py
#  Start node bluerov2_node.py
rosrun bluerov2 bluerov2_node.py
#  Start node bluerov2_control.py
rosrun bluerov2 bluerov2_controller.py
#  Start node bluerov2_teleop.py
rosrun bluerov2 bluerov2_teleop.py

#  Launch files
#  Launch opencm904.launch
roslaunch bluerov2 opencm904.launch
#  Launch record.launch
roslaunch bluerov2 record.launch
#  Launch identification.launch
roslaunch bluerov2 identification.launch
#  Launch bluerov2.launch
roslaunch bluerov2 bluerov2.launch
#  Launch bluerov2_teleop_keyboard.launch
roslaunch bluerov2_teleop_keyboard.launch
#  Launch key_board_velocity.launch
roslaunch bluerov2 key_board_velocity.launch

# Run QGroundControl
# Open new terminal
# Go to the folder of file
cd Downloads
./QGroundControl.AppImage

# Run Plotjuggler
# Open new terminal
plotjuggler-ros
```

```
# Rostopic commands
rostopic bw /topic_name   # display bandwidth used by topic
rostopic delay /topic_name  # display delay for topic which has header
rostopic echo /topic_name  # print messages to screen
rostopic find <msg-type>  # find topics by type
rostopic hz /topic_name    # display publishing rate of topic
rostopic info <topic-name>  # print information about active topic
rostopic list   # print information about active topics
rostopic pub    # publish data to topic
rostopic type  /topic_name   # print topic type
```