Robot Operating System

Content of the labs

The goal of these labs is to discover several of the aspects of using and programming with ROS. There are 4 labs that will deal about increasing complexity of the ROS ecosystem. Most of the labs can be done on your own computer, assuming that you have a working ROS installation.

Setting up the computer

Computers from the lab room at ECN should be already set up. If you want to install ROS on your computer, here are several ways to do so, depending on your motivation:

- Go to ROS website and follow the tutorials on your own.
- Download the Virtual Machine on my webpage. This VM corresponds more or less to the ECN computers and ships with V-REP.
- If you have an Ubuntu, go to my webpage where you can find several helper scripts to install ROS, setup your profile and install V-REP, a simulator that is used for some labs.

At ECN, we assume the `ROS` workspace is located at `~/ros` . Most of the official tutorials use `~/catkin_ws` or something similar but we think a simple `ros` folder is quite explicit. The directory tree is thus:

```
home
 ├── Documents
 ├── Downloads
 └── ros
      ├── build
      ├── devel
      └── src
           ├── some_package
           │    └── CMakeLists.txt
           └── some_other_package
                └── CMakeLists.txt
```

If you decide to use another workspace path, adapt the command lines accordingly.

How to compile

The main tool used to compile ROS packages is called `catkin` . Basically, calling `catkin build` will crawl through all your packages and resolve the dependency tree. It will then compile first the independant packages, then the others. The actual compilation is done through `cmake` and `make` but `catkin` use them automatically.

You have to call `catkin build` each time you have a new package, either because you downloaded one in your workspace or because you manually created one.

How to use QtCreator with ROS

As we saw during C++ labs, an IDE such as Qt Creator is pretty useful. For ROS it will be even more as ROS uses many classes and the IDE will give you autocompletion and hints for their methods. A guide on how to use Qt Creator with ROS is available on my webpage.

Do not forget to launch Qt Creator from the command line (`qtcreator`) or use the desktop shortcut that simulates a console launch.

Once `catkin build` is called, remember that the `build` directory of a given package called `package_name` is at `~/ros/build/package_name` .

ECN computers and the Virtual Machine are configured to compile in Debug mode by default, which helps the bug hunt.

Downloading the labs and ECN ROS package

Packages are downloaded through the `git` tool. They have to be downloaded in the `~/ros/src` directory or sub-directory. To do so, launch a console and go to this directory: `cd ~/ros/src` .

The packages to be downloaded are all in a single git repository, including this pdf file `git clone git@github.com:oKermorgant/ecn_ros_labs.git` You should have the following tree afterwards:

```
home
└─ros
  ├─build
  ├─devel
  └─src
    └─ecn_ros_labs
```

Now go to the ROS workspace or any sub-directory of `src` and type: `catkin build` . You can then use Qt Creator (which is useless for the 1st lab).

In order to run a Python node, the file should be executable (`chmod a+x file.py` in the console, or through the file properties in a folder view).