

Hacking the Deep Learning Robot (to work)

Technical Overview

Minimal Kokubi Installation

In order to control the kokubi base from within the laptop, one only needs the minimal kokubi installation, as portrayed and explained on this page:

<http://yujinrobot.github.io/kobuki/doxygen/enInstallationLinuxGuide.html>

However, if you're on Ubuntu 14.04, you might find it a bit tedious, since there are some things missing in your operating system.

Now I have updated the official tutorial to work with Ubuntu 14.04 as following:

1. install the required libraries:

```
sudo apt-get install python-pip libftdi-dev cmake python-empy python-nose python-  
setuptools build-essential  
sudo pip install wstool catkin-pkg
```

2. now compile from source as follows:

```
mkdir /opt/kobuki_core  
wstool init -j5 /opt/kobuki_core/src  
https://raw.githubusercontent.com/yujinrobot/kobuki_core/hydro/kobuki_core.rosinstall  
cd /opt/kobuki_core  
export PATH=/opt/kobuki_core/src/catkin/bin:${PATH}  
catkin_make  
cd build; make install
```

3. finally, test your installation, by connecting to kokubi base, then running the following commands on your ubuntu computer:

```
echo "export LD_LIBRARY_PATH=/opt/kobuki_core/install/lib" >> ~/.bashrc  
source ~/.bashrc  
sudo su  
/opt/kobuki_core/install/lib/kobuki_driver/demo_kobuki_initialisation
```

this should make an initialization sound come from within kobuki.

this has worked for me, however, when I try to run the simple loop test, the robot doesn't respond so far:

```
/opt/kobuki_core/install/lib/kobuki_driver/demo_kobuki_simple_loop
```

ROS Indigo on Ubuntu 14.04

Now, my second try was installing ROS indigo, which also had some hoops which i had to overcome:

first of all, the "ros-indigo-desktop-full" did not install at all, too many dependency conflicts as of December 2016.

Instead, I went on installing it piece by piece:

1. add the package source list:

```
sudo sh -c 'cat /etc/lsb-release && echo "deb  
http://packages.ros.org/ros.informatik.uni-freiburg.de/ros/ubuntu $DISTRIB_CODENAME  
main" > /etc/apt/sources.list.d/ros-latest.list'
```

2. add the authentication keys for the packages:

```
sudo apt-key adv --keyserver hkp://ha.pool.sks-keyservers.net:80 --recv-key  
421C365BD9FF1F717815A3895523BAEEB01FA116
```

3. install opengl and ecl to avoid conflicts:

```
sudo apt-get install libgl1-mesa-dev-lts-utopic ecl
```

4. update the package list:

```
sudo apt-get update
```

5. install the base ROS system:

```
sudo apt-get install ros-indigo-ros-base
```

6. initialize rosdep:

```
sudo rosdep init  
rosdep update
```

7. find out available packages:

```
apt-cache search ros-indigo
```

8. install the kobuki package:

```
sudo apt-get install ros-indigo-kobuki ros-indigo-kobuki-core
```

9. setup environment variables:

```
echo "source /opt/ros/indigo/setup.bash" >> ~/.bashrc  
source ~/.bashrc
```

10. setup rosinstall:

```
sudo apt-get install python-roscpp
```

11. Set udev Rule:

```
roscpp kobuki_ftdi create_udev_rules
```

logout

unplug the usb cable

login

replug the usb cable

now, let's try a testing package for kobuki, called "keyboard operation":

in a new terminal launch kobuki node:

```
roslaunch kobuki_node minimal.launch
```

in yet another new terminal as well, launch the keyboard operation module:

```
roslaunch kobuki_keyop keyop.launch
```

Warning: a small press can make the kobuki base gain a relatively insane speed