Lab on Selected Topics in Intelligent Robotics – DEIS DT8007

Supervisor: Martin Cooney 2017/9/21 (~ 2 hours)

Today we will do something hands-on with robotics and image processing!

Goal: by completing this lab, the goal is that you will, in designing embedded and intelligent systems:

- Have an idea of how to do basic robotics tasks, like being able to ssh into a remote computer, write a program to a microcontroller, and send messages between different machines (robots and computers).
- Have an idea of how to access images from your raspberry pi's camera, toward applying some common image processing functions to solve problems

Approach: in order to complete the lab you will require the following:

- Adequate preparation. You should have downloaded this file for the lab from Blackboard.
- Facilities. Access to your group's raspberry pi computer (hereafter "rpi") with Robot Operating System (ROS) and an Image Processing library (OpenCV), and a rpi camera. You should use your adapter to power the rpi if you can, not a battery, and disconnect it from the arduino/robot. You will also need to be able to work with the rpi somehow (i.e., you will need to have a computer which can ssh into the rpi; openssh on linux can do this, on windows you can ssh using putty, on android JuiceSSH can do this). There are also apparently some hdmi cables on E1 which can be used to directly hook up your rpis to a monitor. Access as a group to a computer with Arduino installed on it is required; you can also download it on your rpi. You also will want to be able to access as a group the internet with some computer, a project pc or your own computer, to be able to see tutorials. You will work together in your project groups today. For networks: You can use Martin's router (robot3 network I think) for your rpis. You can use your robot's board as an Arduino (otherwise Martin might have a few old Arduinos which could also be tried).
- A submitted report. After the lab, please submit a short report (in pdf format) by October 22 to Martin (marcoo@hh.se), including code, calculations, and/or conclusions.

Result: Your report will be judged as passing, reasonable but failing, or unsubmitted.

Please note:

- If the report is reasonable but failing, you will get a chance to review and resubmit.
- All labs must be submitted to receive a passing grade in the course.
- You are expected to do original work. Using someone else's code/report/etc without referencing them is a _serious_ problem which can result in a failing grade for the course (and possibly more).

Your tasks for the lab:

1. Basic robotics: connecting remotely to a different computer (~10 minutes)

Use ssh to connect to your rpi: e.g., on Linux:

```
sudo ssh pi@192.168.1.X
```

You can find out your rpi's IP address from Martin if you are using robot3 (it will show up in the router as an attached device).

On windows you can ssh easily using Putty (download from somewhere like http://www.chiark.greenend.org.uk/~sgtatham/putty/download.html and run)

On linux it is likely you will be able to ssh just by running the command up above. If your computer says ssh is not installed you may have to install an ssh client using sudo apt-get install ___ or such

On android cell phones, JuiceSSH can be used for ssh.

Take a screenshot and add it to your report.

2. Basic robotics: simple programming of a microcontroller (~10 minutes)

Write a test sketch to your Arduino, then take a screenshot of the Arduino doing something, and add it to your report.

3 Basic robotics: sending messages between machines (~60 minutes)

You will set up your rpi to be able to talk to a different machine via ROS.

a) First make yourself a catkin workspace and package.

http://wiki.ros.org/catkin/Tutorials/create_a_workspace

http://wiki.ros.org/ROS/Tutorials/CreatingPackage

This will involve something like:

```
mkdir -p ~/catkin_ws/src
cd ~/catkin_ws/
catkin_make
source devel/setup.bash
echo $ROS_PACKAGE_PATH
catkin_create_pkg beginner_tutorials std_msgs rospy roscpp
```

b) Follow the tutorial on wiki.ros.org to put a talker and listener program in your source folder (choose to use Python or C++), set up your catkin_make files, and run catkin_make.

Code and instructions can be found in:

http://wiki.ros.org/ROS/Tutorials/WritingPublisherSubscriber%28python%29 (Python)

http://wiki.ros.org/ROS/Tutorials/WritingPublisherSubscriber%28c%2B%2B%29 (C++)

c) Communicate.

Set up your rpi to talk to itself.

Find out your rpi's ip address by typing

```
ifconfig
```

Change your ROS variables in .bashrc

```
sudo nano .bashrc
```

e.g. assuming your rpi's address is 192.168.1.3, make sure it says at the bottom:

```
ROS_MASTER_URI=http://192.168.1.3:11311
ROS_HOSTNAME=192.168.1.3
```

to save your changes:

ctrl-x y return

use `bash --login` in any terminal you are using, or close and reopen

Follow the instructions in the tutorial to run roscore, your talker, and your listener:

http://wiki.ros.org/ROS/Tutorials/ExaminingPublisherSubscriber

Take a screenshot or direct the console output to a file and add to your report.

(If you finish this during labtime you can also use your program to talk to another computer or rpi. You probably also will need to give your rpi a name, and add the other computer to your /etc/hosts file:

```
sudo nano /etc/hosts
```

put its IP address and name in the list, then save changes with ctrl-x y return

Also add your rpi's name to /etc/hosts on the other computer.

Remember only one computer should be ROS master. Make sure the info in .bashrc is correct.

Take a screenshot or direct the console output to a file and add to your report.)

4 Basic image processing: using camera (~10 minutes)

Follow online instructions to attach your camera and enable it (should be something like below).

Open the connector (loosen the top), slide the camera cable in in the correct orientation (on one of mine the blue side of the cable faces the ethernet cable slot), close the connector.

sudo raspi-config -> Interfacing Options -> Enable camera, reboot

Take a photo with your camera

```
raspistill –o test.jpg
```

Move the file from the rpi to your computer

You can do this with a USB drive, or a program like Filezilla, or for linux, you can move to a nice directory on your computer then:

sudo scp pi@192.168.1.X:/home/pi/test.jpg .

Add the image to your reports.

Some other optional things to try:

-If you use VNC you will be able to see what you are doing on the raspberry pi.

Some instructions (note: assumes Raspian but you have Ubuntu):

https://www.raspberrypi.org/documentation/remote-access/vnc/

You can install a client on your own computer, then see xwin on your rpi.

-You can do simple things like opening a video stream or image file, or examples in the opencv folder.

-EXTRA INTERNET RESOURCES ON ROS (for anyone interested)

http://www.ros.org/

https://www.clearpathrobotics.com/assets/guides/ros/index.html