Robotik: Getting started with NAO robot

Dr.-Ing. John Nassour

October 20, 2015

1 The Goal

The goal of this task is to familiarize yourself with robot and run different examples that involves different modules, e.i.: examples for motor control, sensors measurement, and image. Once you get familiarize with the robot, you build your own demo that involves different sensory modalities, motor control, and vision.

2 First Steps

2.1 Account

- Start the Computer: Use Ubuntu 12.04 in the Login Manager
- Use your TU-Chemnitz-Account
- Your $HOME directory (folder) is the same as in the pools of URZ. So, you can access your own documents.
- Check the keyboard type: English, German, etc.

2.2 Access the NAO

- Connect the NAO to the PC with a network cable using ETH1 interface.
- Test the connection (in a terminal) with this instruction: `$ ping nao.local` (Stop with CTRL-C).
- Connect to the NAO:
  `$ ssh nao@nao.local` (Password: nao)
  `nao[*] nao stop` (This command stops running naoqi in the robot)
  `nao[*] nao start` (This command starts running naoqi in the robot)

- You can also start running naoqi in verbose mode as:
  `nao[*] nao stop` (This command stops running naoqi in the robot)
  `nao[*] naoqi -v` (This command starts running naoqi in the robot)
  You can end the program with CTRL-C, this will stop `naoqi`

'NAOqi' is the name of the main software that runs on the robot and controls it. It runs on the robot, under OpenNAO distribution. It can also run on your computer in order to test your code on a simulated robot. For more information about 'NAOqi' please visit this link: `http://doc.aldebaran.com/1-14/dev/tools/naoqi.html`

2.3 Prepare Your NAO

- Connect the NAO to the PC with a network cable using ETH1 interface.
- Test the connection (in a terminal): `$ ping nao.local` (Stop with CTRL-C)
- Now, you need to verify that no user's program is running on your robot. This step is very important to protect your robot and run your program on it. In this step you have to comment by `#` all paths for libraries or Python codes in `autoload.ini`.
- Connect to the NAO:
  `$ ssh nao@nao.local` (Password: nao)
  `nao[*] nao stop` (Stop naoqi)
  `nao[*] nano naoqi/preferences/autoload.ini` (Open `autoload.ini`)
  Comment all paths by adding `#` before the path
  Example:

  ```
  [users]
  #/home/nao/naoqi/blabla1.so
  [python]
  #/home/nao/naoqi/blabla2.py
  ```

  Save changes with CTRL-O
  exit the editor with CTRL-X (now you back to the command line in nao)
  `nao[*] naoqi -v` (please verify if there is any error message from naoqi)
  Your robot is ready now.

2.4 Setup Your Working Environment

This section needs to be done only once. After setting up your working environment for the fist time, you do not need to repeat this step any more.

Create your working directory on the robot, all files of your group should only be inside this directory. Note that if you need to repeat this step when you use a robot for the first time. Connect to the NAO:
`$ ssh nao@nao.local` (Password: nao)
`$ mkdir PraktikumGroup*` (Create a directory for the work and replace the star in the name by your group label. Example: PraktikumGroupA for group A)

Please do not store any file outside your working directory!

3 Start with Examples

Make sure you have a robot ready to use. Make sure Python and Python SDK are installed on your computer. This can be done by opening a terminal (ALT+CTRL+T).

```
$ python2.7
>>> import naoqi
```

If you get no error message, this means that Python SDK are installed and available for python.

3.1 Hello World

1. Start your favorite editor.
2. Open a new window.
3. Copy and paste the following code:

   ```
   from naoqi import ALProxy
   tts = ALProxy("ALTextToSpeech", "<IP of your robot>", 9559)
   tts.say("Hello, world!")
   ```

4. Replace `"<IP of your robot>"` by the IP of your robot. If you don't know its IP address, press its Chest button, NAO will say it. You can also use 'nao.local' as an IP.
5. Save the file as a Python file (myHelloWorld.py).
6. Run it. You can run a python file from a terminal by:

   ```
   $ python2.7 <filename.py>
   ```

As a result your robot says "Hello, world!".

To know how it works, please visit this link: http://doc.aldebaran.com/2-1/getting_started/helloworld_python.html

What you have learned:
To make the robot do something, you have to:
1. Import the module ALProxy.
2. Create an object giving access to one of the NAOqi modules.
3. Call one of its available methods.

3.2 Parallel Tasks - Making NAO Move and Speak

Make sure the robot in a correct sitting position with free space for walking in front of it.

Please read the task description from this link: http://doc.aldebaran.com/2-1/dev/python/making_nao_move.html

Now you are able of setting NAO stiffness, making NAO move, and making NAO move and speak at the same time.

3.3 Getting Sensors Values

Force Sensitive Resistors

What are FSRs? http://doc.aldebaran.com/2-1/family/robots/fsr_robot.html
Run the example: `sensors_getFsrValues.py` (http://doc.aldebaran.com/2-1/_downloads/sensors_getFsrValues.py).
Note that you need to pass the robot IP as a parameter to the main function by:

```
$python2.7 sensors_getFsrValues.py nao.local
```

Inertial Sensor Values

The Inertial unit is made of a 3-axis gyrometers and a 3-axis accelerometer. It is located in the torso with its own processor. For ore information about it please visit this link http://doc.aldebaran.com/2-1/family/robots/inertial_robot.html
Run the example: `sensors_getInertialValues.py` (http://doc.aldebaran.com/2-1/_downloads/sensors_getInertialValues.py).
Note that you need to pass the robot IP as a parameter to the main function by:

```
$python2.7 sensors_getInertialValues.py nao.local
```

3.4 Poses

This section shows how to make NAO go to poses Pose Init and Pose Zero. Please look to predefined postures through this link: `http://doc.aldebaran.com/2-1/family/robots/postures_robot.html`.

Pose Init

Make NAO go to a good initial position.
Run the example: `almotion_poseInit.py` (`http://doc.aldebaran.com/2-1/_downloads/almotion_poseInit.py`).
Note that you need to pass the robot IP as a parameter to the main function by:

```
$python2.7 almotion_poseInit.py --ip nao.local
```

Pose Zero

Put all NAO motors to zero.
Run the example: `almotion_poseZero.py` (`http://doc.aldebaran.com/2-1/_downloads/almotion_poseZero.py`).
Note that you need to pass the robot IP as a parameter to the main function by:

```
$python2.7 almotion_poseZero.py --ip nao.local
```

3.5 Retrieving Images

This section contains examples showing how to get images from NAO's cameras, and how to visualize them.

Get an Image

This example gets an image on the robot. Run the example: `videoInput_getImage.py` (`http://doc.aldebaran.com/2-1/_downloads/videoInput_getImage.py`).
Note that NAOqi's IP address was already set in the python file.

```
$python2.7 videoInput_getImage.py
```

Visualize an Image Using Python Imaging Library

Get one image from NAO, then display it using PIL.
Run the example: `vision_getandsaveimage.py` (`http://doc.aldebaran.com/2-1/_downloads/vision_getandsaveimage.py`).

3.6 Move Joint

1. Start your favorite editor.

2. Open a new window.

3. Copy and paste the following code:

   ```
   import time
   from naoqi import ALProxy

   posture = ALProxy("ALRobotPosture","nao.local",9559)
   mov = ALProxy("ALMotion","nao.local",9559)

   posture.goToPosture("Crouch",0.8)
   mov.rest()

   joint=["LHipPitch","LHipYawPitch","RHipPitch","RHipYawPitch",
   "LAnklePitch","RAnklePitch"]
   mov.setStiffnesses("Body",0.8)
   time.sleep(0.02)

   mov.setAngles("HeadYaw",0.2,1.0)
   mov.changeAngles("HeadYaw",0.2,1.0)

   time.sleep(1.0)
   mov.setStiffnesses(joint,0.0)

   mov.setAngles("RHand",0.0,1.0)
   time.sleep(1.0)
   print mov.getAngles("RHand",True)
   ```

5. Save the file as a Python file (myMoveJoint.py).

6. Run it. You can run a python file from a terminal by:

   ```
   $ python2.7 <filename.py>
   ```

As a result your robot will go to crouch position first and move its head.

4 Try Other Examples

You can find more examples in this link: http://doc.aldebaran.com/2-1/dev/python/examples.html.

5 Your Task

Prepare your own demonstration that involves different sensory modalities and motor control. Make sure that your code is clearly commented.

6 Results to Submit

- A PDF report containing description, comments, and screenshot of each subtask if needed (Please use the template: `https://www.tu-chemnitz.de/informatik/KI/edu/praktikum-robotik/ws2014-15/Report_Template.doc`).

- Source code.

- Note: Comment your code. Also, write your names at the beginning of the source code file.

- Make a demo video showing the results.

- Compress all the required results into a .zip or .tar.gz file.

- Naming convention: LastName1_LastName2_GroupNo_TaskNo.
  Example: If two students from Group "Y" Franz Müller and Peter Maier work together on Task 2, then they submit the file Müller_Maier_GY_T2.zip
  or
  Müller_Maier_GY_T2.tar.gz.
  Don't send the video file if it is too big. You can send a link on it [max size 20 MB].

  Submit the file containing your results before the deadline to: john.nassour@informatik.tu-chemnitz.de.

7 Deadline

Nov. 15th 2015