Using Dynamixel Servos with Python

2.12 Introduction to Robotics – Updated for Fall 2014

By Daniel J. Gonzalez – dgonz@mit.edu

(Please email me if there are any issues)

Overview

This guide is meant to facilitate the use of Python with the Dynamixel MX series Servos. The following steps will allow the user to command the servos to perform sinusoidal control of the 2-DOF robot.

It is expected that the reader know the basics of Python, and is using either Windows or Linux. The library can work on Mac OSX, but the specifics of how to install the libraries in OSX are *not* included in this document. It is also expected that the reader has familiarized him/herself with the functions and capabilities of the Dynamixel servos, via the Robotis website and the motor guide posted on Stellar.

The library being used is a version of the PiRobot pydynamixel library that has been modified to allow for torque control and to provide additional feedback. Note this is NOT “Dynamixel Python Library” that was posted on Staller on October 1st. For the original project, library, and documentation visit https://code.google.com/p/pydynamixel/ and http://www.pirobot.org/.

Procedure:

1. Download and install the RoboPlus software, with included Dynamixel Wizard in order to get the proper drivers.

    1.1. http://www.robotis.com/xe/?mid=download_en&category=20561&document_srl=1132559

    1.2. Note there may be an issue with your FT232 driver for the USB2Dynamixel Dongle. Please check the following website to fix the issue: http://learn.trossenrobotics.com/34-blog/140-ftdi-2-12-00-notice-robotis-usb2dynamixel-cm-530-and-ln-101-october-2014.html#driverchoice

2. Install the pydynamixel Library (Included in 2.12_Python_Dynamixel_Lib.zip)

    2.1. Bring up the command line/terminal

    2.1.1. <u>Windows 7:</u> search for and run “cmd.exe”.

    2.1.2. <u>Ubuntu Linux:</u> CTRL+ALT+T brings up a terminal.

    2.2. Use the **cd** (“Change Directory”) command to browse to the extracted folder.
    PROTIP: Use the TAB button to autocomplete a directory! For example, “C:\U *TAB*” automatically becomes “C:\Users\”. Keep typing and tabbing to select the next directory.

    2.2.1. <u>Windows 7 :</u> **cd C:/Users/*Username*/Desktop/dynamixel-installer**

    2.2.2. <u>Ubuntu Linux:</u> **cd ~/Downloads/dynamixel-installer**

    2.3. Run “setup.py install”

    2.3.1. <u>Windows 7:</u> **python setup.py install**
    Check the output to see if the install was successful

    2.3.1.1. Note: Python may be installed, but running the above command may yield a response of *“'python' is not recognized as an internal or external command, operable*

program or batch file." If that is the case, run the command **dir C:\py*** to determine the name of the python install folder. For example, if on your computer it's called Python27, you can then install the library by running
C:\Python27\python setup.py install

2.3.2. Ubuntu Linux: **sudo python setup.py install**
Check the output to see if the install was successful

2.3.2.1. Note: We want to be running Python 2. Depending on how Python is installed, Python 3 may be the default version of python. Test this by running **python** and checking the version number. You may need to specify **python2** or **python27** instead of **python** in the above command.

3. Download and Install the pyserial Library:

3.1. Ubuntu Linux: from the command line, run **sudo apt-get install pyserial**

3.2. Windows 7: Download and extract "pyserial.win32.exe" from https://pypi.python.org/pypi/pyserial

3.2.1. Follow steps 1.2 through 1.4 above, but for the "pyserial-2.7" folder

4. Test the dynamixel library:

4.1. There are three example files in the …/2.12_Python_Dynamixel_Lib/examples folder. The first two (example1.py and example2.py) show basic position control of 2 servos. The final, LabSinusoidExamplePositionControl.py, performs the same as the proportional controller (servo torque output) with sinusoidal input you wrote in Labview/Matlab in the Oct 17/18 lab.

4.2. To test an example:

4.2.1. Open it in IDLE or another editor to ensure the proper serial port is chosen (COMx in Windows or /dev/ttyUSBx in Linux).

4.2.2. Specify the number of servos in the chain (There are 2 by default).

4.2.3. Plug in the USB2Dynamixel adapter.

4.2.4. Power on the servos to 12V (no more than 14V).

4.2.5. Ensure no other program is using the Serial port (like the Dynamixel Wizard)

4.2.6. Run the example (**F5** in IDLE, **python2 example1.py** in Linux, double click the file in Windows)

4.2.7. Your servos should be moving happily!
If not, something strange is going on (the gains are optimized for the weights and lengths of the current 2-DOF setup. If you remove these weights and MOIs, the current gains may be too high for the system dynamics).