

## Nicu tutorial

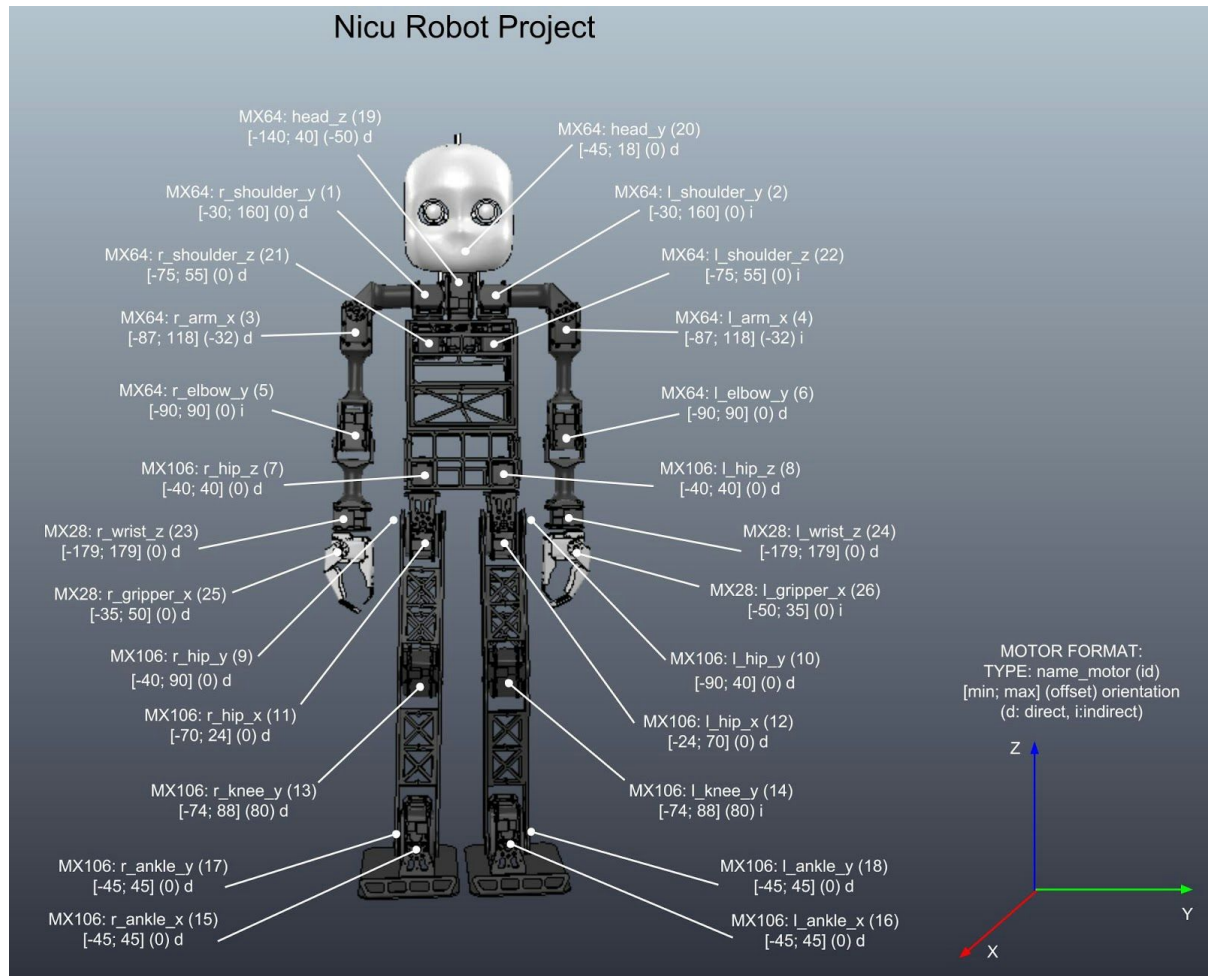
### V-REP

Virtual Robot Experimentation Platform

from Coppelia Robotics (<http://www.coppeliarobotics.com/>)

Last version 3.2.2 released on Sept 5th 2015

### Motors:



### Importing the URDF model:

Find the URDF and the meshes in our git repository:

rzssh1:/informatik/isr/wtm/repositories/git-central/nimbro\_cad.git

in: nimbro\_cad/URDF\_Iris/2012\_08\_12\_NICU

Before you can start working with the URDF you need to set the correct package path by replacing "package://iris\_sandbox//2015\_08\_12\_NICU/ass\_torso\_iam\_fb80b350.stl" in the URDF with the folder that contains all meshes.

## Check 'joint name' and 'axis xyz' before to export the model to V-REP, e.g.:

```
<joint name="r_arm_x" type="revolute">
  <parent link="ass_right_upper_arm:11"/>
  <child link="ass_right_lower_arm:11"/>
  <origin rpy="-0.558505 0 0" xyz="0.0245 -0.073664 -0.022852"/>
  <axis xyz="-1 0 0"/>
  <limit effort="30" lower="-1.518436" upper="2.059489" velocity="1"/>
</joint>
---
<joint name="right_upper_arm" type="revolute">
  <parent link="ass_right_upper_arm:11"/>
  <child link="ass_right_lower_arm:11"/>
  <origin rpy="-0.034907 0 0" xyz="0.0245 -0.073664 -0.022852"/>
  <axis xyz="1 0 0"/>
  <limit effort="30" lower="-0.942478" upper="1.22173" velocity="1"/>
</joint>
```

In V-REP: Plugins -> URDF import (this takes some time)

It is useful to add an associated customization script, for instance, to change the color automatically, like this:

```
...
if (sim_call_type==sim_customizationscriptcall_initialization) then
  -- this is called just after this script was created (or reinitialized)
  -- Do some initialization here

  -- By default we disable customization script execution during simulation, in order
  -- to run simulations faster:

simSetScriptAttribute(sim_handle_self,sim_customizationscriptattribute_activeduringsimulation,false)

simSetShapeColor(sim_handle_all,nil,sim_colorcomponent_ambient_diffuse,{0.25,0.25,0.25})

  floorHandle=simGetObjectHandle('ResizableFloor_5_25_visibleElement')
  simSetShapeColor(floorHandle,nil,0,{0.87,0.87,0.87})

  headHandle=simGetObjectHandle('ass_head_11_visual')
  simSetShapeColor(headHandle,nil,0,{1,1,1})

  leftGripperFixHandle=simGetObjectHandle('ass_left_gripper_MIR_11_visual')
  simSetShapeColor(leftGripperFixHandle,nil,0,{0.75,0.75,0.75})
  leftGripperMoveHandle=simGetObjectHandle('ass_left_gripper_move_11_visual')
  simSetShapeColor(leftGripperMoveHandle,nil,0,{0.75,0.75,0.75})

  rightGripperFixHandle=simGetObjectHandle('ass_right_gripper_11_visual')
  simSetShapeColor(rightGripperFixHandle,nil,0,{0.75,0.75,0.75})
  rightGripperMoveHandle=simGetObjectHandle('ass_right_gripper_move_11_visual')
  simSetShapeColor(rightGripperMoveHandle,nil,0,{0.75,0.75,0.75})

end
...
```

Or using the V-REP scene with the nicu there

### Remote API control:

It is possible to control the robot only with v-rep functions, but then the behavior is not replicated in the real robot.

### PyPot control

First, we need to install the robot control library, have a look into:

<https://poppy-project.github.io/pypot/installation.html>

Check out the pypot documentation here: <http://poppy-project.github.io/pypot/>

If you want to see an example to control a Poppy robot, follow the following tutorial:

How-To: Control a Poppy Humanoid in a Simulator using a Python lib: pypot

<http://nbviewer.ipython.org/github/poppy-project/poppy-humanoid/blob/master/software/samples/notebooks/Controlling%20a%20Poppy%20humanoid%20in%20V-REP%20using%20pypot.ipynb>

### Have a look in our wiki:

[https://www2.informatik.uni-hamburg.de/WTM/wtm/wtmwiki/index.php/NICU\\_Working\\_with\\_Poppy\\_Framework](https://www2.informatik.uni-hamburg.de/WTM/wtm/wtmwiki/index.php/NICU_Working_with_Poppy_Framework)

### Control simulated robot with pypot:

```
import pypot.vrep

pypot.vrep.close_all_connections()

nicu = pypot.vrep.from_vrep('nicu_humanoid_only_upper_tmp.json', '127.0.0.1', 19997,
    '../vrep/nicu_color.ttt')

nicu.r_arm_x.compliant=False
nicu.r_arm_x.goal_speed=5
nicu.r_arm_x.goal_position=90
raw_input("Press enter to continue...")
```

### Or:

```
#import pypot.robot
#from pypot.robot import from_json
import pypot.vrep

def move_joint(joint, speed, position):
    setattr(eval('nicu.'+joint), 'compliant', False)
    setattr(eval('nicu.'+joint), 'goal_speed', speed)
    setattr(eval('nicu.'+joint), 'goal_position', position)

pypot.vrep.close_all_connections()
nicu = pypot.vrep.from_vrep('nicu_humanoid_only_upper_tmp.json', '127.0.0.1', 19997,
    '../vrep/nicu_color.ttt')

#nicu = from_json('nicu_humanoid_only_upper_tmp.json')

joint = 'r_arm_x'
move_joint(joint, 5, 90)
raw_input("Press enter to continue...")
```

**Connect with the real robot:**

ssh nimbro@134.100.10.165

password: nimbro

\* go to the folder: cd pyPotControl/

\* switch the servos on: python switchServosOn.py

\* and run your code.....

**Future work:**

- A new version of URDF model
- To add vision sensor to the Nicu robot in the simulator
- Create script to simplify pure shapes
- ...