## MATLAB SETUP TO WORK WITH CASPROS

**àFor Matlab files: Use CASPR\_private shared by Gino**

CASPR Directory structure

Timeline

Description automatically generated

**àImportant file locations**

ROSInterfaceàCASPR\_MYO\_ROSInterface/CASPR\_private/src/HardwareInterface/CableRobotActuator/MYO-muscle/ MyoInterface.m

SimulationàCASPR\_private\_scripts\members\Dominic\OperationalSpaceControl

HardwareàCASPR\_private\_scripts\members\Dominic\Myo-Hardware

*Actual implementation of the controllers are inside the following classes*

**àClasses**

CASPR\_private\master\src\Analysis\Control\OperationalSpaceControl\ReactiveQPControl

CASPR\_private\_scripts\members\Dominic\OperationalSpaceControl\IterativeLearningControl

**àFile modification for making the CASPR work CASPROS**

Following modifications are made in the CASPR\_private library

1. CASPR\_MYO\_ROSInterface/CASPR\_private/scripts/local/CASPR\_private\_scripts/members/Dominic/Useful Scripts/CreateSphericalTraj.m à Create the spherical trajecoty.
2. In script\_BMArm\_.... make the following changes
   1. Comment
      1. % CASPR\_configuration.SetGlobalModelMode(ModelModeType.COMPILED);
      2. % CASPR\_configuration.SetReuseCompiled(1);
   2. Change
      1. cdpr = model\_config.getModel('WORKING', model\_config.defaultOperationalSetId);

to

cdpr = model\_config.getModel('WORKING', model\_config.defaultOperationalSetId,ModelModeType.COMPILED);

* + 1. traj\_filename = sprintf('%s/scripts/local/%s.otraj', CASPR\_configuration.LoadHomePath(), trajectory\_id);

to

traj\_filename = sprintf('%s/%s.otraj', CASPR\_configuration.LoadHomePath(), trajectory\_id);

**[[1]](#footnote-2)**

## IMPORTANT GIT DIFFERENCE

The modifications from the last part are also shown by ‘git diff <commit\_previous> <commit\_present> <file\_path>’

|  |
| --- |
| **11/06/2020: Git ‘difference’ showing the changes made by Dominic for running the BMARm** |
| script\_BMArm\_ILC\_MyoExperiment.m |
|  |
| Text  Description automatically generated |
|  |
| script\_BMArm\_MyoExperiment.m |
|  |
| Text  Description automatically generated |
|  |

|  |
| --- |
| **26/06/2020: Git ‘difference’showing the changes made by Dominic for running the BMARm** |
| script\_BMArm\_MyoExperiment.m |
| Text  Description automatically generated |
| script\_BMArm\_ILC\_MyoExperiment.m |
| Text  Description automatically generated |
| MyoExperiment.m |
| Text  Description automatically generated |

## HOW TO RUN CASPROS WITH FPGA:

**Preparation**

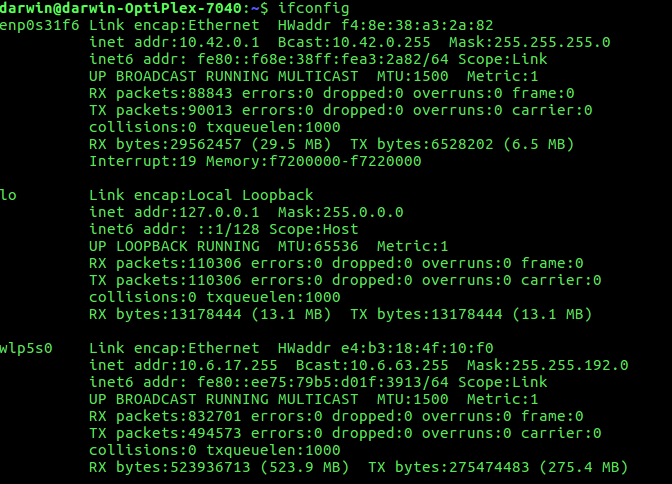
1. Connect the Ethernet cable
2. Check the ethernet cable connection with arp -a:

Graphical user interface, text, chat or text message

Description automatically generated

1. Start ‘roscore’ on the desktop of Linux computer.
2. Perform ROS Ethernet configuration setup on the PC:
3. Modify .bashrc on Linux PC:
4. Change the ROS\_IP to 10.42.0.1
5. The IP can be different according to the ifconfig:

(might need to source the .bashrc file for activating the changes)





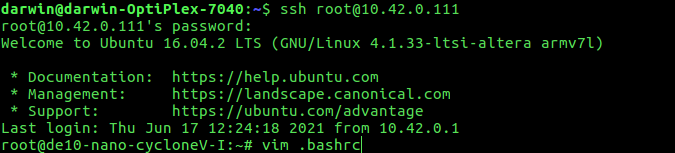
1. ‘ssh’ to nano board because linux is also running on it.

|  |
| --- |
| ssh root@10.42.0.111 |
| Password: Roboy2016 |

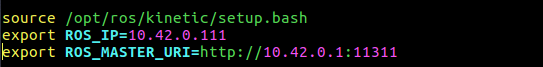
1. Perform ROS Ethernet configuration setup on the nano:
2. Modify .bashrc on FPGA board:

|  |
| --- |
| vim .bashrc |

(might need to source the .bashrc file for activating the changes)



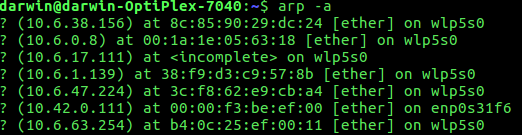
1. Modify the ROS\_IP accordingly (The same method as modifying the ip in Linux PC):



ROS\_IP should be the IP address of the DE nano board.

**Hardware Steps**

1. Connect the Ethernet cable
2. Check the ethernet cable connection with arp -a:



1. Connect to the FPGA board with the ip address in arp -a (Terminal 1):

|  |
| --- |
| ssh root@rosrun casper |
| Password: Roboy2016 |

1. Open a new ternimal (Terminal 2) on Linux PC and do:

|  |
| --- |
| roscore |

1. Start the roboy\_plexus in the DE nano terminal (Terminal 1, ssh one) before powering on the robot motors

|  |
| --- |
| ./roboy\_plexus |

1. Power on the motors
2. Open another terminal window (Terminal 3), source the catkin package.

|  |
| --- |
| source ~/caspros/devel/setup.bash |

1. Turn on the myo\_manager (Terminal 3 ) to get access to the FPGA board:

|  |
| --- |
| roslaunch myo\_manager launch\_BMArm\_with\_FPGA.launch |



1. Open another terminal window (Terminal 4), source the catkin package.

|  |
| --- |
| source ~/caspros/devel/setup.bash |

1. Run (Terminal 4 ) the low-tension node to tighten the BMArm

|  |
| --- |
| rosrun caspr\_cpp low\_tension |



Running matlab script (not done):

All the devices should be connected to a switch, and we need to check the ip addresses and modify all the three .bashrc

Open another terminal window, source the catkin package.

source ~/caspros/devel/setup.bash

We will be running the matlab interface node:

rosrun caspr\_cpp matlab\_to\_myomuscles



**Precautions:**

|  |
| --- |
| 1. Cut the switch (red circular) if its on and then supply power. |
|  |
| Note: Noise form the power supply fan is normal |
|  |
| 1. Twist the switch counter clockwise and the power lights (first light) on motor will turn on green. |
|  |
| 1. rosrun caspr\_cpp low\_tension |
|  |
| 1. stop myo\_manger, roboy\_plexus, other terminals running then power the motors off. |
|  |
| 1. If required cables can be dragged out, if its really tight then the cables are stuck in the motor pulleys. |
|  |

Change the cables

Cables are long enough but not too long enough

A picture containing indoor

Description automatically generated

Text

Description automatically generated

## HOW TO CONNECT A LINUX PC AND WINDOWS PC THROUGH ETHERNET CABLE FOR ROS

Connection between Linux PC and Any Windows Laptop:

1. Connect both of the computer with ethernet cables to a switch.
2. Do ifconfig on Linux and ipconfig on windows, and find the IP address for both of them. They should be like 10.42.0.1 on Linux and 10.42.0.X on windows.
3. Try to ping the Linux PC on windows machine with:

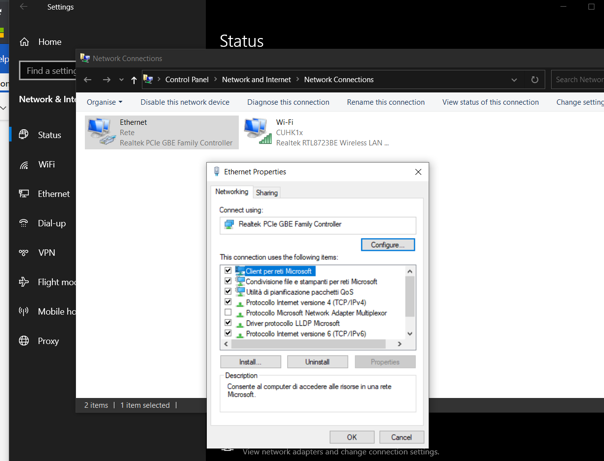
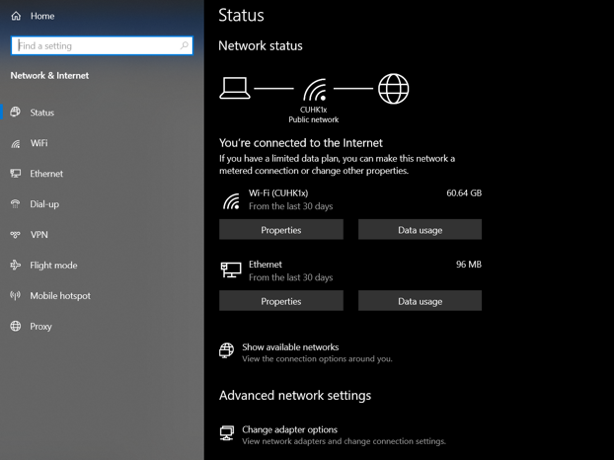
|  |
| --- |
| ping 10.42.0.1 |

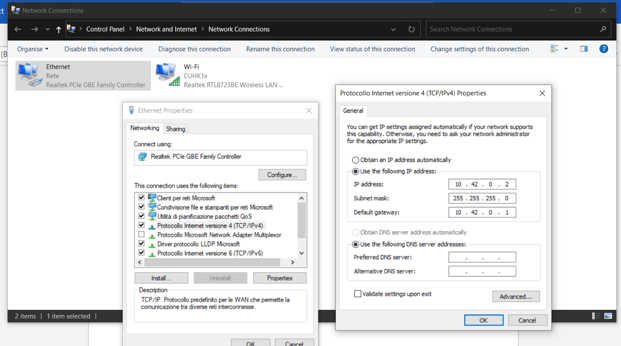
If not succeed, check the IP address of the Linux machine and the ethernet connection.

1. Try to ping the window machine on the Linux PC with:

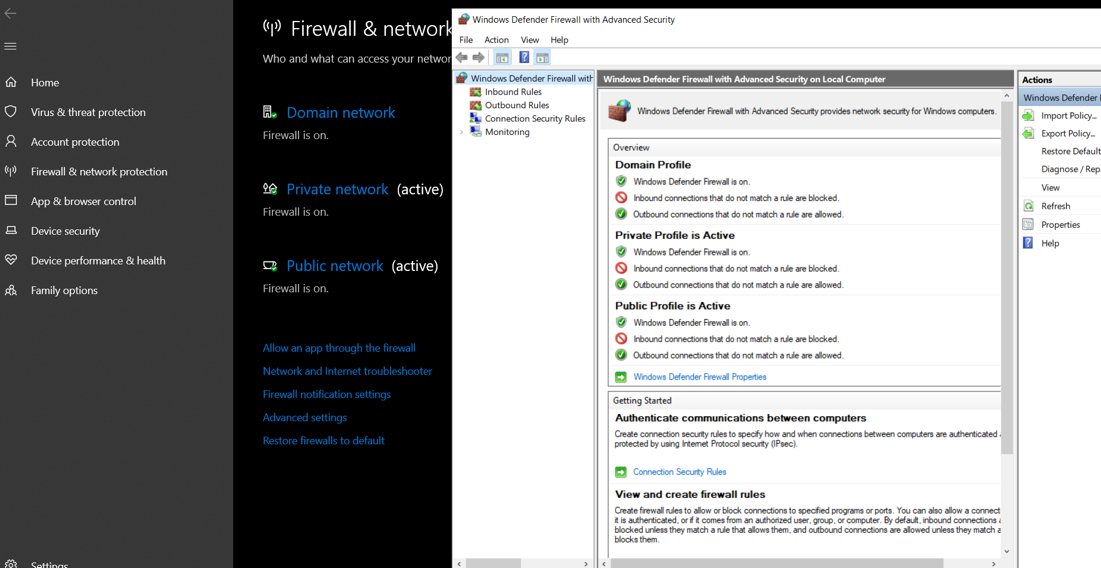
|  |
| --- |
| ping 10.42.0.X (which is the IP address of the windows machine) |

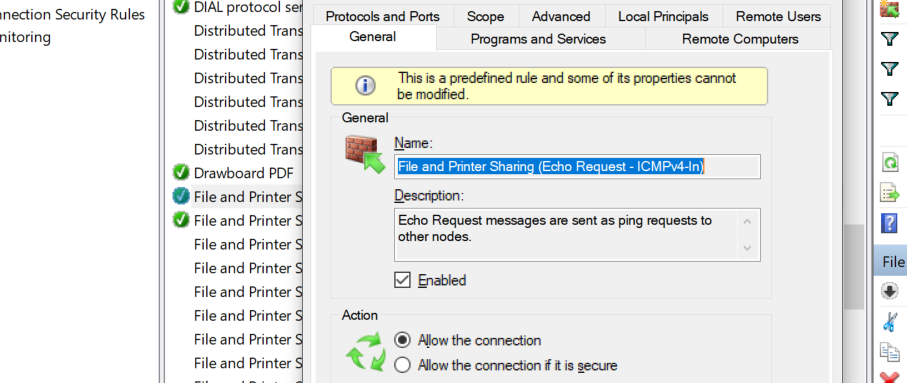
If not working, we need to change some settings of the windows firewall as following:



` Graphical user interface, text, application

Description automatically generated```````````````````





## HOW TO CONNECT WINDOWS MATLAB PC AND ROS RUNNING ON LINUX PC

1. Run roscore
2. Ssh to the FPGA and ./roboy\_plexus
3. Roslaunch myo\_manager
4. Rosrun myo\_muscle (Follow the steps described in ‘Hardware Steps’)
5. Run matlab on windows

Caution: Ros not initialized

Check the ROS\_MASTER\_URI 10.42.0.1 (The ROS\_MASTER\_URI should be same as that of the Linux PC, where ROS master is running)

Type rosinit in the matlab command line window to check if the ROS toolbox is installed

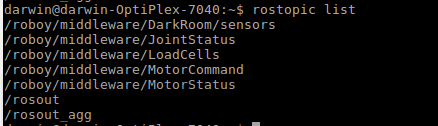
*Can not create subscriber, check the rostopic list on the Linux PC, and rerun the steps above to make sure /MATLAB/FeedBack is on the list*

Rostopic reference:

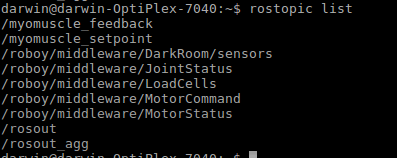
Only roscore:

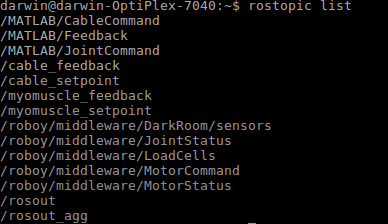


Add plexus:



Add myo\_manager:



Add rosrun caspr\_cpp matlab\_to\_myomuscle:

## IMPORTANT NODE-TOPIC DIAGRAMS FROM RQT

./roboy\_plexus

Diagram

Description automatically generated

rosrun caspr\_cpp matlab\_to\_myomusclesDiagram

Description automatically generated

./roboy\_plexus

roslaunch myo\_manager launch\_BMArm\_with\_FPGA.launch

rosrun capr\_cpp matlab\_

Diagram

Description automatically generated



## QUESTIONS ASKED DURING MEETING WITH DOMINIC

26/06/2020

* Important packages/nodes
  + *low\_tension* :Tighten the cables
  + *rest*: hold length commands and position
  + *matlab\_to\_myomuscles*
  + *position\_control\_xml* :Test the BMArm’s joint angles by running the arm in vertical, horizontal, twisting and flexing motions. IT sends commands from *CASPR\_ROS*.
* Why the FK of cables to joints is in the *CASPR\_ROS?*
  + Due to time constraint Dominic could not implement it in the MATLAB.
* MyoExperiment 146, what is the usage of *f\_feedback?*
  + To check whether the BMArm is actuated by the positive force in the reactive controller
  + ILC
* What’s the input and output interface of BMArm (FPGA Board ./roboy\_plexus)?
  + Roboy plexus is to control the myomuscle and let it go to the certain position.
* How to set the initial position of the cables?
  + *low\_tension* initial position to rest
  + *rest*: hold length command and position
* How to do the cable length calibration, test the FK?
  + Not much information or idea.
* Check the understanding of the flow chart
  + The flow chart drawn on Inkscape is correct
* Why the second part of FK (joint to task) is inside the controller?
* The second simulation experiment with unexpected traj is not match
  + Could not fix it
* ILC error function
* The choice of *alpha and beta*
* The stability

## IMPORTANT CONCLUSIONS REACHED

* Since Darwin has already implemented the *ControllerSImulator* class in CASPR, thus, Dominic separated the simulations and hardware. He executed simulations directly with Darwin CASPR framework and implemented the hardware class, *MyoExperiment* class separately.
* By increasing time span between the joint space trajectory points, the OL controller performed well. We have also reduced the time step from 0.01 s to 0.033 s.

1. |  |
   | --- |
   | \* Chen: He is not an author and even he don’t knows about the MATLAB implementation thoroughly. |
   | \*\*Author: 1st Dominic aka Yin Pok Chan (Masters), 2nd author is another master student who left. |

   [↑](#footnote-ref-2)