sEMG Acquisition & co-contraction level online Computation Procedure

Armando Amerì

July 2022

Contents

1 Device

1. Switch on the device.
2. The LED starts to blink.
3. Connect the Dongle (Bluetooth Receiver) to the laptop.
4. Launch the file 'Start_OYMotion_New.bat' → 'New_gForceEMGArmband_0042\OTrain_v0.7.1'.
5. Wait for COM Enabling and save the COM number shown in the Terminal (Figure 1) .
6. Connect from OTrain Application (Figure 2).
7. Bracialet's LED stops to blink.
8. Close the Terminal and the Application.

2 MATLAB & SIMULINK

1. Open the following files:
   - 'gForce_data_acquisition_ccOnline_ROS_v2_1.slx'
   - 'fnc_to_be_implemented.m'
2. enstablish a correct port connection between the simulink machine and the ROS master
3. in Matlab Command Window launch the command shown in 5.1 with the correct rosmaster ip
4. in 'gForce_data_acquisitionn_ccOnline_ROS_v2_1.slx' set the COM port saved in point Device.5 (Figure 3)
5. in 'gForce_data_acquisitionn_ccOnline_ROS_v2_1.slx' [Acquisition Data] (Figure 4) → 'Bracialet' set the correct bracialet you are using
6. in 'gForce_data_acquisitionn_ccOnline_ROS_v2_1.slx' [Acquisition Data] → 'Data Selection' set value to be evaluated: Raw data, rotation matrix, ...
7. in 'gForce_data_acquisitionn_ccOnline_ROS_v2_1.slx' [Acquisition Data] → 'Acquisition Phase' set: Acquisition
8. in 'gForce_data_acquisitionn_ccOnline_ROS_v2_1.slx' [Parameter's Settings] set parameters (at the moment the only tested are 1000 Hz,8 bit,8ch)
9. in 'gForce_data_acquisitionn_ccOnline_ROS_v2_1.slx' set the Manual Switch 'up' and the Switch Button 'off' (Figures 5 and 6)
10. in 'gForce_data_acquisitionn_ccOnline_ROS_v2_1.slx' [ROS Communication] set to 'off' all the Radio Buttons (Figures 7)
11. in 'gForce_data_acquisitionn_ccOnline_ROS_v2_1.slx' Settings and for all block set the right sample time 1/f
12. in fnc_to_be_implemented.m run [Inizialization] section (see matlab_code 5.2)
13. run file 'gForce_data_acquisitionn_ccOnline_ROS_v2_1.slx'
14. The LED starts to blink (Data send)
15. Wait for initial peak in channels' scope
16. Start to move arm and hand to acquire data (move as in paper [1])
17. Stop file 'gForce_data_acquisitionn_ccOnline_ROS_v2_1.slx' (Please note: the bracialet continue to send data)

at this point let's make the bracialet stop sending data:

1. in 'gForce_data_acquisition_ccOnline_ROS_v2_1.slx'[Acquisition Data] set 'Acquisition Phase': Stop
2. run file 'gForce_data_acquisition_ccOnline_ROS_v2_1.slx'
3. Bracialet's LED stops to blink
4. Stop file 'gForce_data_acquisition_ccOnline_ROS_v2_1.slx'

Please Note: all datas acquired in this session at point 13 are saved on 'previousSession' workspace variable in order to avoid cancellation.

continue...

1. in fnc_to_be_implemented.m [Plot] section take the range of data of the entire willing signal and of the stiffness phase (see matlab_code 5.3 and Fig. Acquisition_1)
2. in fnc_to_be_implemented.m [Range] section (see matlab_code 5.4) set the range selected in point 15, as descripted in the comment, so run the section
3. in fnc_to_be_implemented.m run [Computation] section
4. in 'gForce_data_acquisition_ccOnline_ROS_v2_1.slx' [Acquisition Data] set 'Acquisition Phase': Acquisition
5. in 'gForce_data_acquisitionn_ccOnline_ROS_v2_1.slx' set the Manual Switch 'down' and the Switch Button 'on' (Figures 5 and 6)
6. in 'gForce_data_acquisitionn_ccOnline_ROS_v2_1.slx' [ROS Communication] set to 'on' the required Radio Buttons (Figures 7)
7. run file 'gForce_data_acquisitionn_ccOnline_ROS_v2_1.slx' to acquire the cc level online

3 ROS

1. Switch on the device.
2. The LED starts to blink.
3. In Terminal 1 run the command "roslaunch ros_gforce calibration.launch"
4. In Terminal 2 run the command "rostopic echo −p /datas3 > $PATH_OF_ROS_GFORCE/ros_gforce/csv/$CSV_NAME.csv"
5. Start doing the calibration.
6. When the calibration is finished, close Terminal 1 and 2
7. In Terminal 1 run the command "rosrun ros_gforce import_file_x_plot.py $CSV_NAME"

8. Select the 2 ranges [extreme_left_total_signal extreme_right_total_signal extreme_left_stiff extreme_right_stiff]

9. Close the figures and Terminal 1.

10. In Terminal 1 run the command "rosrun ros_gforce import_file_x_matrix_2.py $MATRIX_FILE_NAME"

11. In Terminal 1 run the command "rosrun ros_gforce emg_signal_processing_full _channel_notch_high_maslide_Online_Header_2.py $MATRIX_FILE_NAME"

4 Pictures

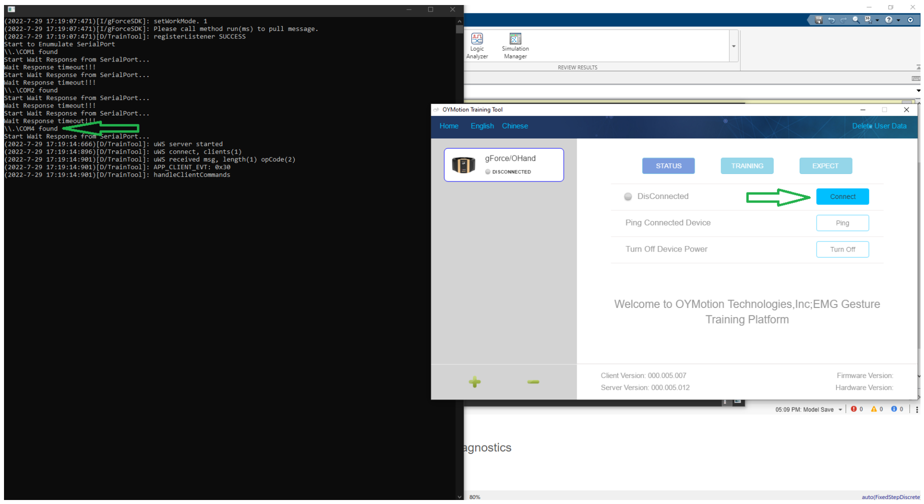

Figure 1: Device Connection

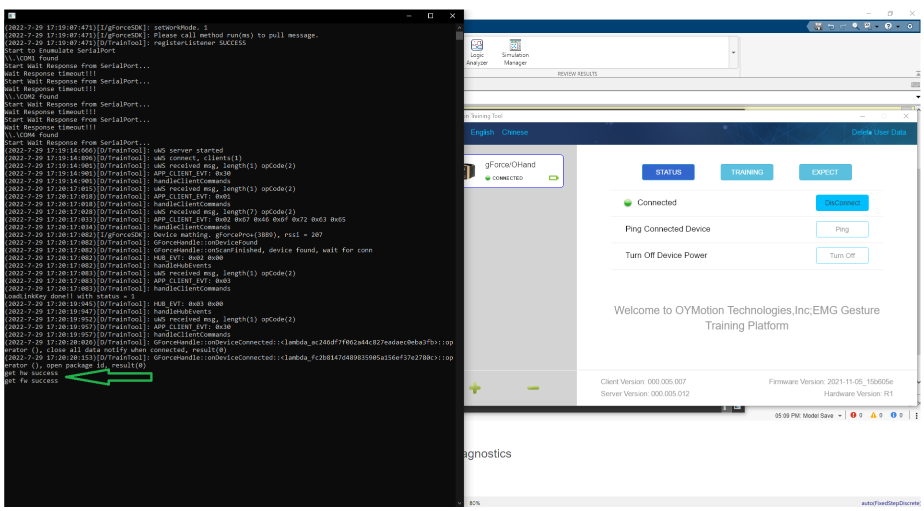

Figure 2: Device Connection

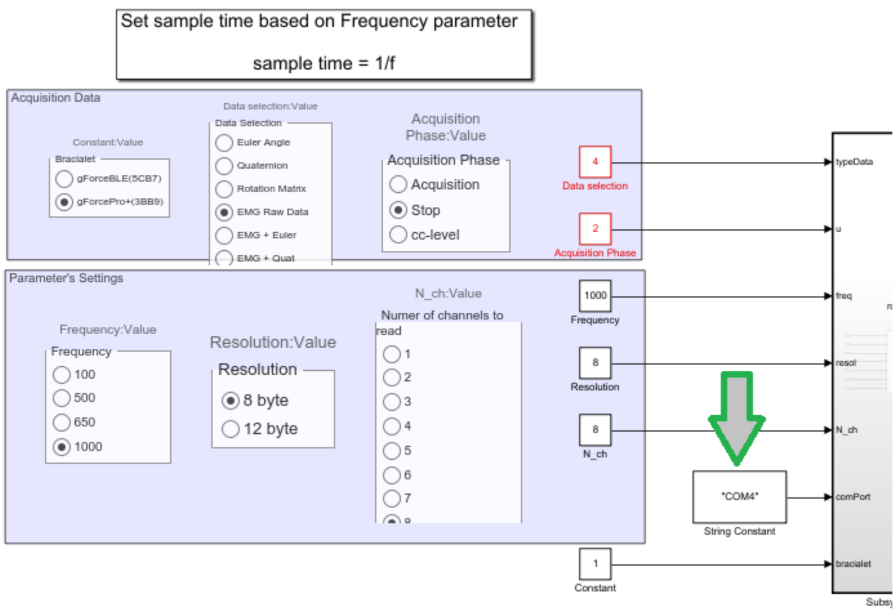

Figure 3: COM Port Setting

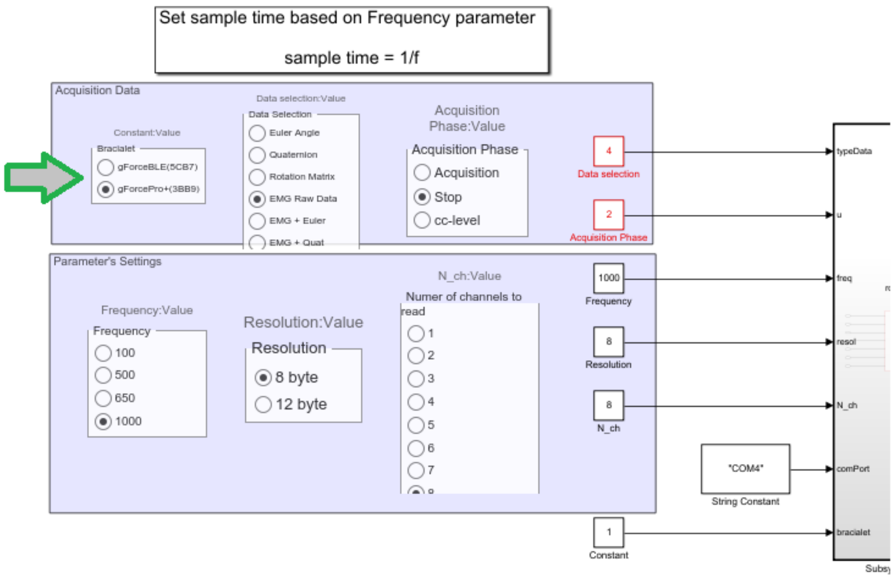

Figure 4: Acquisition Data Setting

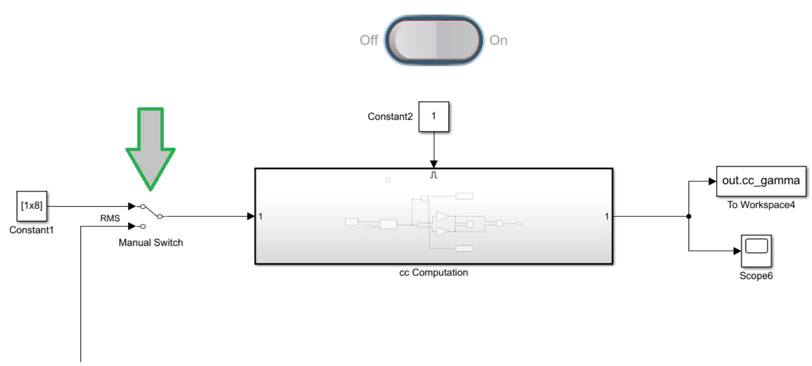

Figure 5: EMG Manual Switch

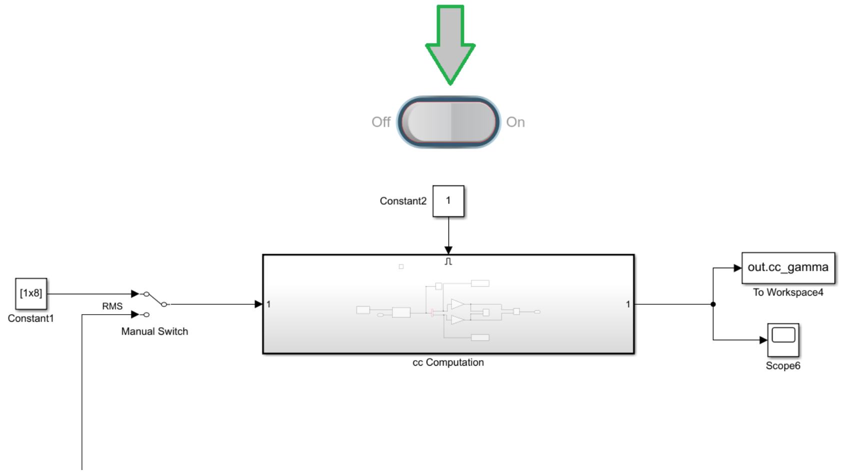

Figure 6: cc Activation

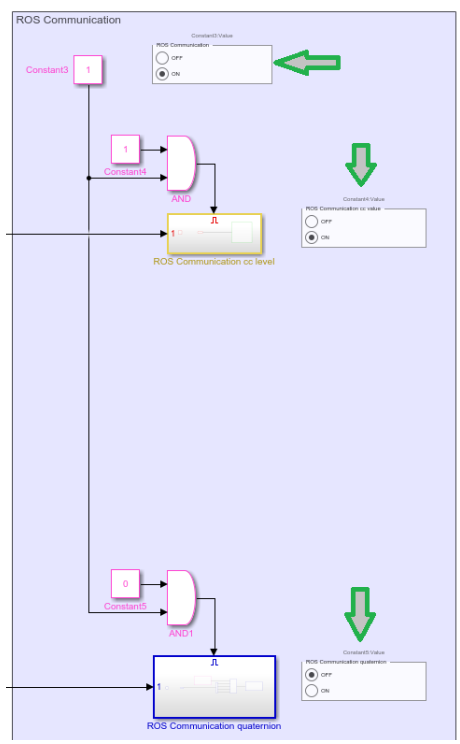

Figure 7: ROS Communication

5 Code

5.1 ROS_init

```
rosinit('192.168.17.128')
```

5.2 Inizialization

```
%% Inizialization
M_pinv = zeros(2,8);
k_ext = 1;
k_flex = 1;
```

5.3 Plot

```
%% Plot
% load('calibration_New');

plot(out.emg_rms.Data)
A = out.emg_rms.Data';
```

5.4 Range

```
%% Range
range = [79642;92017;89500;90100]; % [
    extreme_left_total_signal extreme_right_total_signal
    extreme_left_stiff extreme_right_stiff]
diff_range = range(2,1)-range(1,1);
diff_range_2 = range(4,1)-range(3,1);
```

5.5 Computation

```
%% Computation
[M, U_Offline] = nnmf(A(:,range(1,1):range(2,1)),2);
plot(U_Offline')
M_pinv = pinv(M);

s_ext = M(:,1);
s_flex = M(:,2);

u_ext = U_Offline(1,:)';
u_flex = U_Offline(2,:)';

k_ext = sum(u_ext)/diff_range_2;
k_flex = sum(u_flex)/diff_range_2;
```