

# ***Procedure for Validating an OpenExo System before Use***

## ***Purpose***

This guide assumes you have set up your exoskeleton system according to the steps [here](#), up until the “Conducting the First Trial” section.

The purpose of this guide is to outline the general steps for validating an OpenExo system before use. This is particularly aimed at OpenExo systems which use torque sensors, FSRs, or other sensing devices which inform exoskeleton control. The method used to accomplish this is by using the calibration manager controller to make sure the motor and torque sensor directions are correct. Once the directions are set properly, the motors will be unplugged and the desired controller run to verify that torque profiles and sensor readings look as anticipated without risking unanticipated torque application.

## ***Calibration Manager***

The first step is to run the calibration manager controller to ensure that the direction of torque application is correct, and that, if used, the torque sensors are reporting in the correct direction.

The following is taken from the documentation for the calibration manager controller found in the “Documentation” folder of the OpenExo code.

~~~~~

### **Description**

This controller applies a small, constant torque while printing statements of relevance (command value/sign, torque sensor value/sign, angle sensor value/sign) to the Arduino Serial Monitor.

## **Parameters**

There are no user defined parameters associated with this controller.

## **Purposes**

After you have built an exo, one of the first things to check is the motor directions and the signs (positive/negative) of any of your external sensors (torque, angle, etc.).

Having correct motor directions is important as it allows the exo controller designers to focus on the control itself, knowing that, if calibrated properly before use, the exo assistance will always be in the designed directions.

The same principle applies for the sensors involved with control. Since PID control is implemented in many of our exo controllers, it is crucial to make sure that the directions match.

This article will elaborate how to conduct the direction calibration using the CalibrManager controller.

On each exo, this calibration process only needs to be done **once**, its results will need to be **manually** saved on the SD card and will be loaded **automatically** every time the exo starts up.

## **Motor directions**

When designing a controller, the values returned from the "calc\_motor\_cmd()" function determines the motor torque command (feedforward). To ease the controller design process, we propose that controller designers follow the following direction definitions for torque commands:

- At both ankle joints: Positive for Dorsiflexion, and negative for Plantarflexion
- At both knee joints: Positive for Extension, and negative for Flexion
- At both hip joints: Positive for Flexion, and negative for Extension
- At both shoulder joints: Positive for Flexion, and negative for Extension
- At both elbow joints: Positive for Flexion, and negative for Extension

In summary, this follows the right-hand rule.

## **Torque sensor signs (positive/negative)**

Following the motor direction definitions, we need to verify that, when the motor supplies a positive torque command, the torque sensors, when zeroed properly, also returns a positive reading when the motor shaft is not spinning.

## How to calibrate

1. Set Calibration Manager as the default controller for the current exo.
2. Connect the Teensy to a computer through USB and open up the Arduino Serial monitor.
3. Connect the exo to the python GUI.
4. Power on the exo and start a new session.
5. Make sure you are securely holding the joint of interest (either with your hand or worn on your body) so that it can not freely move. If left unsecured the joint will actuate upon start which could cause injury.
6. Upon start, the Calibration Manager will send a positive torque command (3.5 Nm by default) to each motor.
7. Feel the direction that the torque is trying to actuate.
8. If the torque is not rotating in the positive direction (see above for the specific directions), power off the device, remove the SD Card, and flip the motor direction to its correct direction by modifying the appropriate "JointFlipMotorDir" in "Config.ini".
9. After modifying all motor directions (if needed), start the test over again [once again making sure the device is secured], and this time focus on the sensor readings (e.g., torque, angle) shown on the Arduino Serial Monitor.
10. Again feel the direction that the torque is trying to actuate (it should now match the definitions above).

11. Now try to manually move the joint back to its neutral position (e.g., move the hip upright away from a flexed position into a neutral position).

12. If the corresponding sensor does not return a positive value, you will need to modify the SD card to flip the sensor direction.

13. Again, power off the device, remove the SD card, and flip the sensor direction by modifying the appropriate "JointFlipSensorDir" in "Config.ini".

~~~~~

## ***Verifying Torque Profile***

With the motor/torque sensor directions calibrated, we'll now **unplug the power wires from the motors**, put the exoskeleton on, run the controller that will be used with the system, and visually inspect the torque profile to ensure it looks as anticipated.

First, as mentioned, **unplug the power wires from the motors**. Unplugging the motors allows the controller to run and the GUI to plot as normal without the motors applying torque. Set the active controller in the SD card to the one which you intend to use with the exoskeleton. Be sure to set the PID flag to 0, even if you intend to use PID. Attempting to use PID without the motors providing torque will cause a large discrepancy between the prescribed torque and the reading from the torque sensors. This won't harm anything but will potentially cause the torque profile to look abnormal, confusing the test about to be performed. With the motors unplugged, put the exoskeleton on and power it on. Again, you will need to have set up the exo according to [this guide](#) up until the "Conducting the First Trial" section.

Start the GUI, connect the exoskeleton, and press “Calibrate Torque Sensors.” After a couple seconds, the start trial button will be available to press. Press it and observe the torque profile (some controllers have a built in calibration phase, in which case you will need to walk around to calibrate the device). The torque profile should be smooth (or a regular step profile in the case of the step controller) and should not greatly exceed the torque setpoint defined in the SD card. If the profile is noisy with sudden, large changes in amplitude, this is most often due to faulty sensors (most often FSRs). If the torque profile looks as expected but is either very large or very small in magnitude, this is likely due to the torque setpoint in the SD card being set improperly.

With the preceding steps performed and assuming the torque profile looks as expected (and after ending the trial and turning the device off), you may plug the motors back in and conduct a trial with torque.