Chapter 6

Classroom Exo Quick Start Guide

Summary

This document serves as a quick start quide, to get started with your already assembled Exoskeleton. We will walk you through the necessary MATLAB installation, how to upload the embedded software and how to use the Graphical User Interface (GUI) elements.

6.1 Necessary Software and IDEs

The main software components of the Classroom Exo are the embedded software that controls the Exoskeleton's Hardware, and a Graphical User Interface (GUI) that enables you to interact with all the different functionalities. You will be able to design a pid controller or use the sEMG sensor to control the Motor movement, while adjusting thresholds and other setting easily in MATLAB .Since we are passionate about this Open Source Project, all the files can be accessed on our Github.

To be able to use all the Software you will need to install the following IDEs on you Laptop/PC:

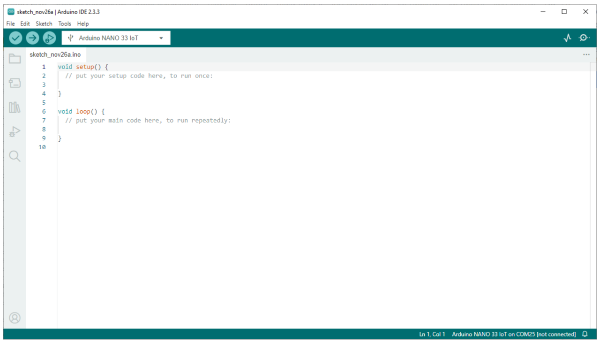
- Arduino IDE
- Mathworks MATLAB (from Version R2023b onward)

Installing the Arduino IDE

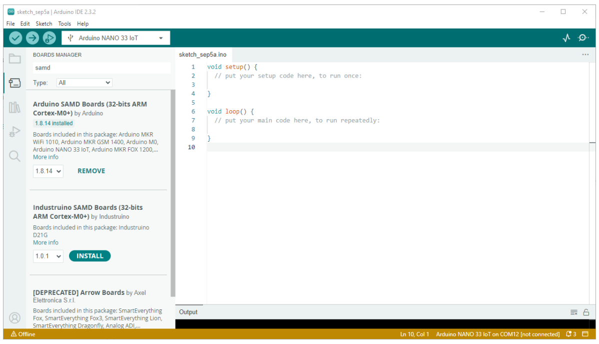
Before you can start programming the microcontroller, you have to install the Arduino Integrated Development Environment (IDE). You can download it for several operating systems on the Arduino website: https://www.arduino.cc/. When you install and open the IDE in Figure 6.1a, you will find everything you need to write code and upload it to the Arduino board. To program the Arduino microcontroller, you have to connect it to your computer with a micro-USB cable. You can refer to this link here to get the files to check the servo and sensor using Arduino.

Hint

For more advanced users we recommend using Visual Studio Code with the PlatformIO IDE extension, which offers more advanced debugging options for more complex programs. Please refer to the PlatformIO setup guide here: https://platformio.org/install/ide?install=vscode



(a) *We use the Arduino IDE for programming all the Components Tests.*



(b) *Make sure to download the correct package for your board.*

Figure 6.1: *Glimpse of the Arduino IDE and the Board manager*

Installing Mathworks MATLAB

Before you can start using the graphical user interface, you will need to install the MATLAB software which is provided by Mathworks. Check with your university to get access to their Mathworks license, create your own Mathworks student account and follow the instructions on their website here: https://matlab.mathworks.com When you install and open the IDEin Figure 6.2, you will find everything you need to write code in MATLAB. For your ease we have developed a MATLAB GUI, to interact with the Exoskeleton more easily. Refer to our Github repository, in order to get access to the GUI.

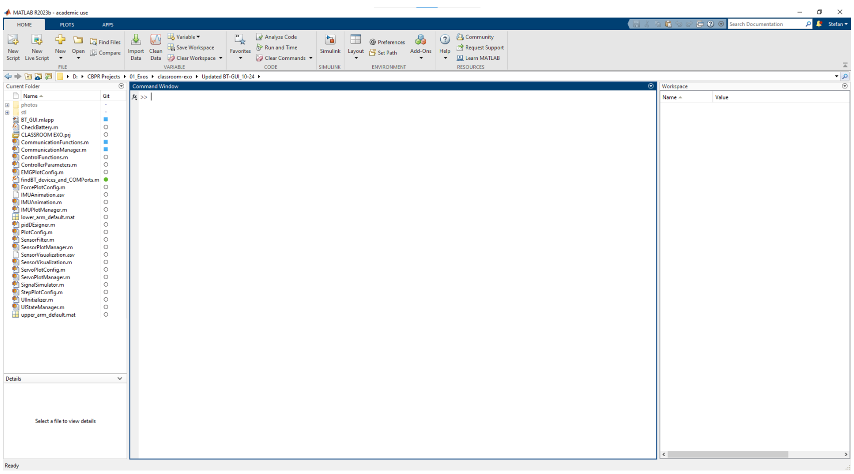


Figure 6.2: *Glimpse of the MATLAB IDE by Mathworks.*

6.2 Embedded Software

Summary

In this section we will walk you through the functionalities in the embedded code that is running directly on the SAMD21 chip of the Arduino Nano 33 IoT. With the embedded code, we enable Bluetooth Classic communication, different control algorithms for the motor, IMU integration as well as battery and device status indications. This makes the Classroom Exo more User friendly, while also offering a compatible GUI in MATLAB to easily change between control modes.

6.2.1 Setting up Bluetooth Classic Communication

The Arduino Nano 33 IoT has two separate chips on its board. The SAMD21 ist the heart of the microcontroller, which is running the main program that handles serial communication, reading the sensors and controlling the servo motor. The second chip, the WiFi NINA 102 module, enables us to use WiFi, Bluetooth BLE or Bluetooth Classic to communicate with the world outside of the microcontroller. The WiFi Nina chip essentially repeats, what the bigger SAMD21 chip sent to it, and passes it through to the outside. This signal can then be received by our MATLAB GUI.

The Classroom Exo should already have Bluetooth enabled, there are two ways to test if the Bluetooth Setup was already completed:

- Check available Bluetooth devices on your mobile phone
- Check available Bluetooth devices in the Bluetooth settings on Windows

If the device shows up on both connection managers, your Classroom Exo is ready to go. If the device does not show up, check that the Battery is charged and refer to our more comprehensive Guide on how to setup the Bluetooth Communication.

6.2.2 Control Algorithms in the embedded Software

Our embedded software enables you to run the Classroom Exo with different control modes, using the signals from the angle, force and sEMG sensor.

Embedded PID Controller

- standard P-I- or D controller using K_p, K_i and K_d,
- standard P-I- or D controller using design and control parameters,

Thresholding Control using Force Sensor The Force Sensor, which is located at the wrist cuff, enables us to implement

- a simple threshold based control,
- an admittance/proportional control

Thresholding Control using sEMG Sensor

- a simple threshold based control with either one sEMG sensor,
- or a simple threshold based control with two sEMG sensors

More information on how the Algorithms work, can be found in the more comprehensive Guide!

6.3 GUI in MATLAB

Summary

This section shows the MATLAB Front-End, which consists of an intuitive Graphical User Interface (GUI). The GUI communicates via Bluetooth Classic with te embedded code on the Arduino. The GUI enables controlling the Classroom Exo with a simulated P-/I-/D-Controller, Thresholding based on the Force or sEMG sensor. The wireless communication enables updating setting, streaming adds safety and freedom to move in space The fast communication also enables a simulatneus 3D animation of the Classroom Exo based on realtime IMU and angle data in the user interface. Additional information, like control mode and battery percentage, are also shown to the user.

6.3.1 Getting your MATLAB ready

Before we can use the GUI, you will need to install MATLAB and some additional Packages. Using the Addon Manager you can install the following packages, if you didn't already do that during your initial MATLAB installation. The following packages are essential for using the complete GUI:

- Control System Toolbox (version 23.2)
- Instrument Control Toolbox (version 23.2)
- Robotics System Toolbox (version 23.2)
- Sensor Fusion and Tracking Toolbox (version 23.2)
- Signal Processing Toolbox (version 23.2)
- Symbolic Math Toolbox (version 23.2)

Hint

The MATLAB GUI was developed in Version R2023b, and tested for both R2023b and R2024a, so we **CAN NOT** guarantee that all the functionalities of the GUI will work with either older or newer versions!

6.3.2 Pairing your Classroom Exo with Windows 10 onward

Hint

Pairing the Bluetooth device was tested on Windows 10 and 11. So we cannot guarantee that it works with other operating systems! As of right now there is no support for MacOSx or Linux, and they are **NOT** planned for future releases!

To use the device, you must pair it with your Laptop/PC:

1. Power the BT device on
2. In Windows go to System Settings → Bluetooth and other devices → Add devices → Bluetooth
3. Now search for new available devices and pair the desired device e.g. *xx_CLASSROOM_EDU_EXO_PRO*.

After pairing the device, you can use it with the MATLAB GUI.

6.3.3 Getting started with the GUI

Open the MATLAB application by double-clicking on the file. Now, the MATLAB app will search for available Bluetooth devices, as you can see in Figure 6.3, that match our device naming structure.

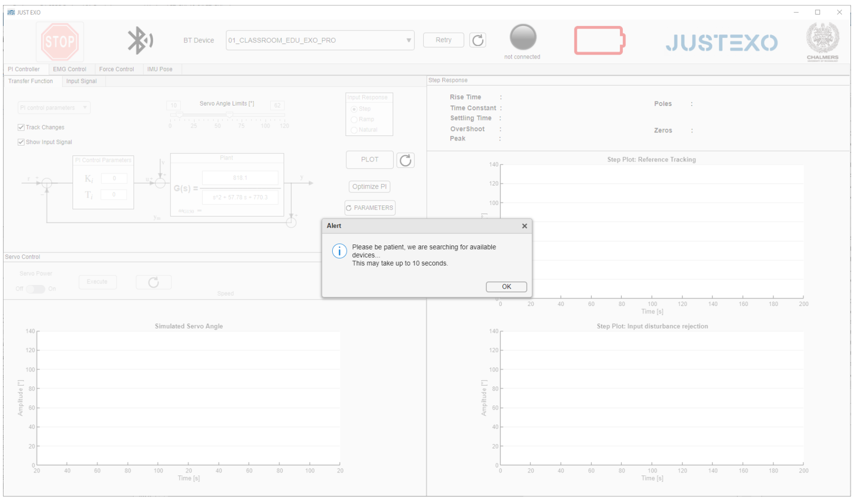


Figure 6.3: *Startup page of the MATLAB GUI searching for available Bluetooth devices.*

From the drop-down menu you are now able to select your desired device, and just hit the *connect* button. When the connection was successful, you will be able to see the updated Bluetooth symbol, the status led and battery icon.

Now, you can select the tab with the control mode that you want to try out.

PID Control Tab

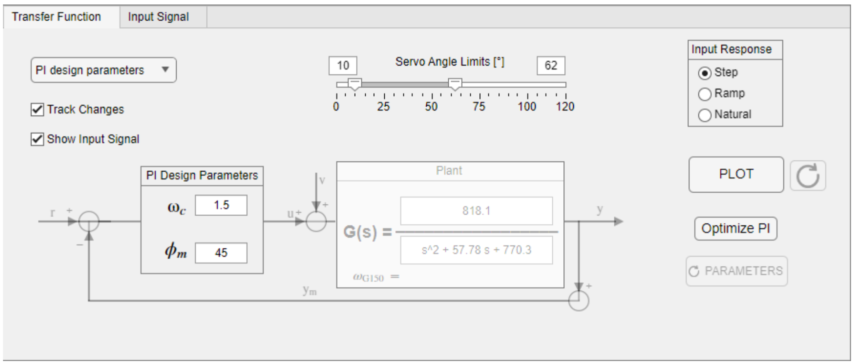


Figure 6.4: *The controller can be designed by different parameters, such as the design parameters ω_c and ϕ_m.*

With the initial values you will have a system response that exceeds safe limits. Now, you should adjust the values so you get the desired control system behaviour.

To speed things up a bit, we've implemented an optimization algorithm in the *Optimize PI* button. As you can see in Figure 6.5 the PI-controller responds a lot slower, and does not exceed the desired limits of the the step input signal.

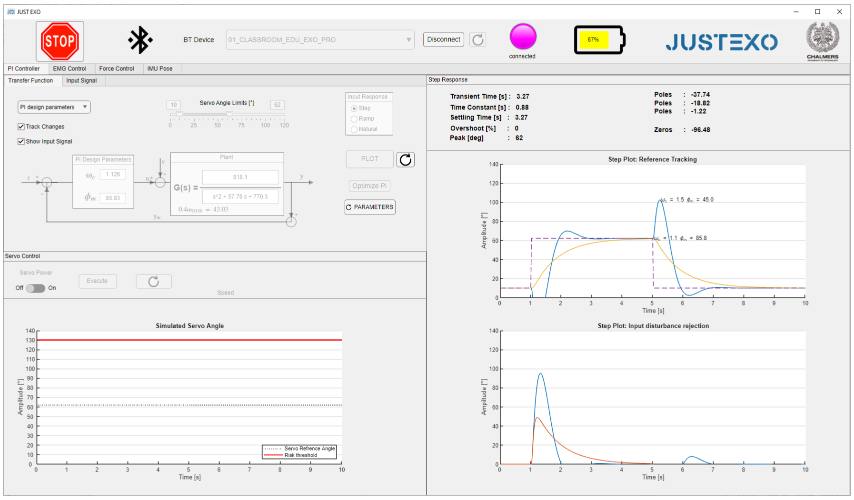


Figure 6.5: *The controller can be optimized by an automatic algorithm.*

Force Control Tab

If you want to use the threshold-based force control, go to the *Force Control Tab*. There a predefined settings for the thresholds and servo motor speed. You can just hit the *Start* button, and try out how the resistance feels against your wrist. You can see *Force Control Tab* interface in Figure 6.6.

You can adjust the force thresholds really easily: change the value on the threshold sliders and hit *Apply Thresholds* - this will restart the plot from zero with the newly selected threshold values.

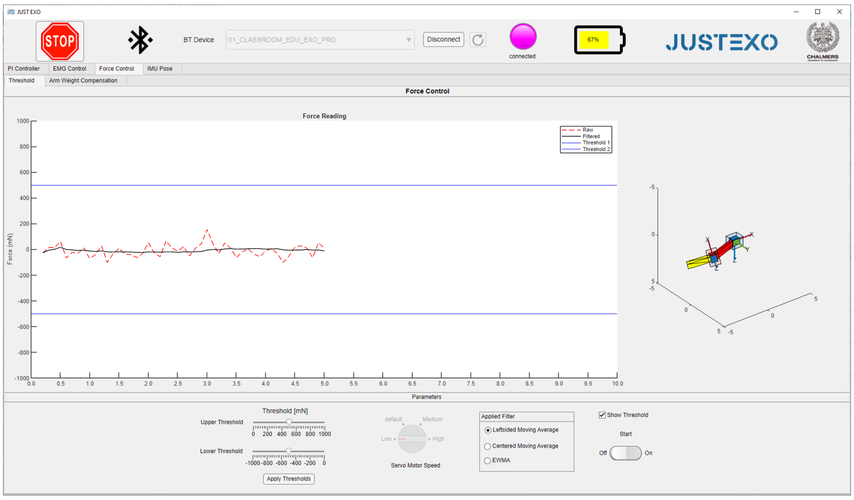


Figure 6.6: *The Force Control Tab interface enables you to control the servo motor by applying pressure to the wrist cuff on the Exoskeleton.*

You can also observe how different filters act on the signal quality. The filters on the radio-buttons are just for display purposes, and have no effect on the motor control itself.

On the right hand side, you can see a 3D-plot that displays a reference of the Exoskeleton based on IMU and angle sensor data. When you turn, or move the upper arm of you Exo, you can observe the movement in the liveplot. When the servomotor moves, you can see how the lower arm moves in reference to the upper arm.

sEMG Control Tab

If you now want to use the sEMG based control, go to the *EMG Control Tab*. You will find predefined settings for the thresholds and servo motor speed, as well as the sEMG channel. You can see *EMG Control Tab* interface in Figure 6.7.

Now, connect your sEMG sensor to a big muscle (i.e. biceps brachii) and plug the cable into the AUX port with the label *EMG 0*. If you want to use the other channel, you also have to change the channel in the drop-down menu!

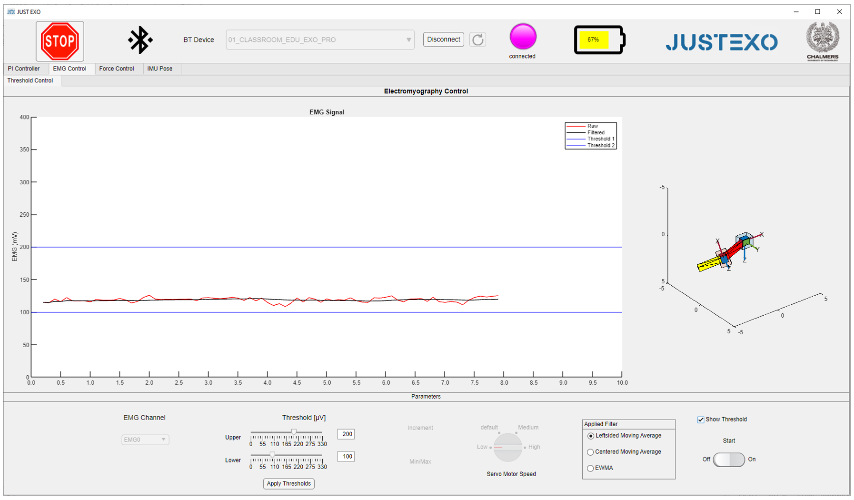


Figure 6.7: *The sEMG Control Tab interface enables you to control the servo motor by flexing and relaxing your muscles.*

Again, you can hit the *Start* button and try to control the servo motor with your muscle activation.

On the right hand side, you will find the same 3D liveplot as in the *Force Control Tab*.

6.3.4 Disconnecting from your device

Before powering off your Exoskeleton it is crucial to disconnect your Matlab GUI properly! Just hit the disconnect button, this will reset the Exoskeletton to it's initial mode - so you will see the led button blinking blue again. Your Matlab interface will reset, and start searching for paired devices again.

If you want to exit your GUI, you can now just close the Matlab application; and also turn off the Exoskeletton manually.

6.3.5 Emergency Stop

If you do not feel comfortable using the Exoskeleton anymore, or a dangerous situation occurs there are two options of shutting down the servo motor movements.

1. Hitting the *Emergency Stop* button in the GUI - this turns off the motor power supply and disconnects the device from your MATLAB GUI. You will need to close the GUI before you can use it again.
2. Hitting the led power button on your Exoskeleton manually - this powers off the entire Exoskeleton without properly disconnecting from your MATLAB GUI. You will need to remove your paired device from the Windows settings, and start from the very beginning of this tutorial!

Acronyms

GUI Graphical User Interface. 1, 3, 4, 5

IDE Integrated Development Environment. 1, 2

sEMG Surface Electromyography. 1, 3, 4